Midi Player Tool Kit for Unity

Thierry Bachmann
Version 2.6
Mon Nov 25 2019

Table of Contents

Table of contents

Namespace Index

Packages

Here are the packages with brief descriptions (if available):

Hierarchical Index

Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Namespace Documentation

MidiPlayerTK Namespace Reference

Classes

- class MidiExternalPlayer
  [MPTK PRO] - Script for the prefab MidiExternalPlayer. See full example TestMidiExternalPlayer.cs with a light sequencer. Play a midi file from a path on the local deskop or from a web site
- class MidiFileLoader
  Script for the prefab MidiFilePlayer. Load a midi file. List of Midi file must be defined with Midi Player Setup (see Unity menu MPTK).
- class MidiFilePlayer
  Play a midi file. Midi files must be defined with Midi Player Setup (see Unity menu MPTK) from Unity editor.
- class MidiFileWriter
  [MPTK PRO] - Write a midi file from differents sources based on NAudio frawemork. See full example TestMidiWriter.cs with a light sequencer.
- class MidiInReader
  Play generated notes. Any Midi file is necessary rather create music from your own algorithm with MPTK_PlayEvent(). Duration can be set in the MPTKEvent, but a note can also be stopped with MPTK_StopEvent().
- class MidiListPlayer
  [MPTK PRO] - Script for the prefab MidiListPlayer. Play a list of pre-selected midi file from the dedicated inspector. List of Midi files must exists in MidiDB. See Midi Player Setup (Unity menu MPTK).
- class MidiLoad

Base class for loading a Midi file. No seqencer, no synthetizer. Usefull to load all the Midi events from a Midi.

- class MidiPlayerGlobal
  Singleton class to manage all global features of MPTK.
- class MidiStreamPlayer
  Play generated notes. Any Midi file is necessary rather create music from your own algorithm with MPTK_PlayEvent(). Duration can be set in the MPTKEvent, but a note can also be stopped with MPTK_StopEvent().
- class MidiSynth
- class MPTKEvent
  Midi Event class for MPTK. Usage to generate Midi Music with MidiStreamPlayer or to read midi events from a Midi file with MidiLoad or to recevice midi events from MidiFilePlayer OnEventNotesMidi.
- class MPTKListItem
  A list of string with index: midi, preset, bank, drum, ...
- class TrackMidiEvent
  Midi event list (NAUdio format)

Enumerations

- enum MPTKCommand : byte { MPTKCommand.NoteOff = 0x80, MPTKCommand.NoteOn = 0x90, MPTKCommand.KeyAfterTouch = 0xA0, MPTKCommand.ControlChange = 0xB0, MPTKCommand.PatchChange = 0xC0, MPTKCommand.ChannelAfterTouch = 0xD0, MPTKCommand.PitchWheelChange = 0xE0, MPTKCommand.Sysex = 0xF0, MPTKCommand.Eox = 0xF7, MPTKCommand.TimingClock = 0xF8, MPTKCommand.StartSequence = 0xFA, MPTKCommand.ContinueSequence = 0xFB, MPTKCommand.StopSequence = 0xFC, MPTKCommand.AutoSensing = 0xFE, MPTKCommand.MetaEvent = 0xFF }
- *MIDI command codes* enum MPTKController : byte { MPTKController.BankSelect = 0, MPTKController.Modulation = 1, MPTKController.BreathController = 2, MPTKController.FootController = 4, MPTKController.MainVolume = 7, MPTKController.Pan = 10, MPTKController.Expression = 11, MPTKController.BankSelectLsb = 32, MPTKController.Sustain = 64, MPTKController.Portamento = 65, MPTKController.Sostenuto = 66, MPTKController.SoftPedal = 67, MPTKController.LegatoFootswitch = 68, MPTKController.ResetAllControllers = 121, MPTKController.AllNotesOff = 123, MPTKController.AllSoundOff = 120 }
- *MidiController enumeration http://www.midi.org/techspecs/midimessages.php#3* enum MPTKMeta : byte { MPTKMeta.TrackSequenceNumber = 0x00, MPTKMeta.TextEvent = 0x01, MPTKMeta.Copyright = 0x02, MPTKMeta.SequenceTrackName = 0x03, MPTKMeta.TrackInstrumentName = 0x04, MPTKMeta.Lyric = 0x05, MPTKMeta.Marker = 0x06, MPTKMeta.CuePoint = 0x07, MPTKMeta.ProgramName = 0x08, MPTKMeta.DeviceName = 0x09, MPTKMeta.MidiChannel = 0x20, MPTKMeta.MidiPort = 0x21, MPTKMeta.EndTrack = 0x2F, MPTKMeta.SetTempo = 0x51, MPTKMeta.SmpteOffset = 0x54, MPTKMeta.TimeSignmature = 0x58, MPTKMeta.KeySignature = 0x59, MPTKMeta.SequencerSpecific = 0x7F }

MIDI MetaEvent Type

Enumeration Type Documentation

enum MidiPlayerTK.MPTKCommand : byte [strong]

MIDI command codes

Enumerator:

NoteOff	Note Off
NoteOn	Note On
KeyAfterTouch	Key After-touch
ControlChange	Control change
PatchChange	Patch change
ChannelAfterTouch	Channel after-touch
PitchWheelChange	Pitch wheel change
Sysex	Sysex message
Eox	Eox (comes at end of a sysex message)
TimingClock	Timing clock (used when synchronization is required)
StartSequence	Start sequence
ContinueSequence	Continue sequence
StopSequence	Stop sequence
AutoSensing	Auto-Sensing
MetaEvent	Meta-event

enum MidiPlayerTK.MPTKController : byte [strong]

MidiController enumeration http://www.midi.org/techspecs/midimessages.php#3

Enumerator:

BankSelect	Bank Select (MSB)
Modulation	Modulation (MSB)
BreathController	Breath Controller
FootController	Foot controller (MSB)
MainVolume	Main volume
Pan	Pan
Expression	Expression
BankSelectLsb	Bank Select LSB ** not implemented **
Sustain	Sustain
Portamento	Portamento On/Off
Sostenuto	Sostenuto On/Off
SoftPedal	Soft Pedal On/Off
LegatoFootswitch	Legato Footswitch

ResetAllControllers	Reset all controllers
AllNotesOff	All notes off
AllSoundOff	All sound off

enum [MidiPlayerTK.MPTKMeta](#) : byte `[strong]`

MIDI MetaEvent Type

Enumerator:

TrackSequenceNumber	Track sequence number
TextEvent	Text event
Copyright	Copyright
SequenceTrackName	Sequence track name
TrackInstrumentName	Track instrument name
Lyric	Lyric
Marker	Marker
CuePoint	Cue point
ProgramName	Program (patch) name

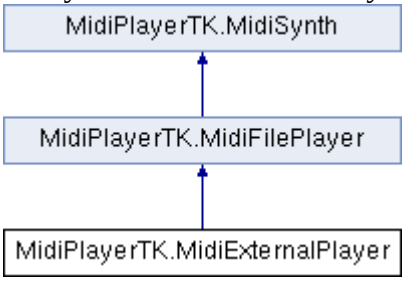
DeviceName	Device (port) name
MidiChannel	MIDI Channel (not official?)
MidiPort	MIDI Port (not official?)
EndTrack	End track
SetTempo	Set tempo
SmpteOffset	SMPTE offset
TimeSignmature	Time signature
KeySignature	Key signature
SequencerSpecific	Sequencer specific

Class Documentation

MidiPlayerTK.MidiExternalPlayer

[MPTK PRO] - Script for the prefab MidiExternalPlayer. See full example TestMidiExternalPlayer.cs with a light sequencer. Play a midi file from a path on the local deskop or from a web site

Inheritance diagram for MidiPlayerTK.MidiExternalPlayer:



Public Member Functions

- new void MPTK_Play ()
  Play the midi file defined in MPTK_MidiName
- new void MPTK_Next ()
  Play next Midi - NO EFFECT for external
- new void MPTK_Previous ()
  Play previous Midi - NO EFFECT for external
- void MPTK_Stop ()
  Stop playing
- void MPTK_RePlay ()
  Restart playing of the current midi file
- void MPTK_Pause (float timeToPauseMS=-1f)
  Pause the current playing
- void MPTK_UnPause ()
  Pause the current playing
- MPTKEvent.EnumLength MPTK_NoteLength (MPTKEvent note)
  Return note length as https://en.wikipedia.org/wiki/Note_value
- MidiLoad MPTK_Load ()
  Load the midi file defined with MPTK_MidiName or MPTK_MidiIndex. It's an optional action before playing a midi file witk MPTK_Play.
- void MPTK_InitSynth (int channelCount=16)
  Init the synthetizer. Prefabs automatically initialize the synthetizer (see events). It's not usefull to call this method.
- void MPTK_ClearAllSound (bool destroyAudioSource=false)
  Clear all sound

Public Attributes

- EventNotesMidiClass OnEventNotesMidi
  Define unity event to trigger when notes available from the Midi file.
- EventStartMidiClass OnEventStartPlayMidi
  Define unity event to trigger at start of playing the Midi.
- EventEndMidiClass OnEventEndPlayMidi
  Define unity event to trigger at end of playing the midi.
- bool MPTK_CorePlayer
  If true then Midi events are read and play from a dedicated thread. If false, MidiSynth will use AudioSource gameobject to play sound. This properties must be defined before running the application from the inspector. The default is true.
- bool MPTK_DirectSendToPlayer
  If true (default) then Midi events are sent automatically to the midi player. Set to false if you want to process events without playing sound. OnEventNotesMidi Unity Event can be used to process each notes.
- bool MPTK_EnableChangeTempo
  Should accept change tempo from Midi Events ?
- bool MPTK_PauseOnDistance
  Should the Midi playing must be paused if distance between AudioListener and MidiFilePlayer is greater than MaxDistance
- bool MPTK_EnablePanChange
  Should change pan from Midi Events or from SoundFont ?
- bool MPTK_EnablePresetDrum

Should accept change Preset for Drum canal 10 ? Disabled by default. Could sometimes create bad sound with midi files not really compliant with the Midi norm.

- bool MPTK_LogWave
  Log for each wave to be played
- uint MPTK_ReleaseTimeMin = 500000
  [Only when CorePlayer=False] Define a minimum release time at noteoff in 100 iem nanoseconds. Default 50 ms is a good tradeoff. Below some unpleasant sound could be heard. Useless when MPTK_CorePlayer is true.
- int MPTK_StatVoiceCountActive
  Count of the active voices (playing) - Readonly
- int MPTK_StatVoiceCountFree
  Count of the free voices for reusing on need. Older than AutoCleanVoiceTime are removed when count is over than AutoCleanVoiceLimit - Readonly
- float MPTK_StatVoiceRatioReused
  Percentage of voice reused during the synth life. 0: any reuse, 100:all voice reused (unattainable, of course!)
- int MPTK_StatVoicePlayed
  Count of voice played since the start of the synth
- int MPTK_AutoCleanVoiceLimit
  Free voices older than MPTK_AutoCleanVoiceLimit are removed when count is over than MPTK_AutoCleanVoiceTime
- bool MPTK_WeakDevice
  Should play on a weak device (cheaper smartphone) ? Apply only with AudioSource mode (MPTK_CorePlayer=False) Playing Midi files with WeakDevice activated could cause some bad interpretation of Midi Event, consequently bad sound.
- EventSynthClass OnEventSynthAwake
  Unity event fired at awake of the synthesizer. Name of the gameobject component is passed as a parameter.
- EventSynthClass OnEventSynthStarted
  Unity event fired at start of the synthesizer. Name of the gameobject component is passed as a parameter.

Properties

- new string MPTK_MidiName `[get, set]`
  Full path to Midi file or URL to play. must start with file:// or http:// or https://.
- new int MPTK_MidiIndex `[get, set]`
  Index Midi to play or playing - NO EFFECT for external
- bool MPTK_PlayOnStart `[get, set]`
  Should the Midi start playing when application starts ?
- bool MPTK_Loop `[get, set]`
  Should automatically restart when Midi reaches the end ?
- double MPTK_Tempo `[get]`
  Get default tempo defined in Midi file or modified with Speed. Return QuarterPerMinuteValue similar to BPM (Beat Per Measure)
- float MPTK_Speed `[get, set]`
  Speed of playing. Between 0.1 (10%) to 5.0 (500%). Set to 1 for normal speed. Be carefull when modifying speed on fly from GUI. Each change generates 0.3s of pause, avoid little and frequent speed change.
- double MPTK_Position `[get, set]`
  Set or Get midi position time from 0 to lenght time of midi playing (in millisecond). No effect if the Midi is not playing.

- bool MPTK_IsPaused `[get]`
  Is Midi file playing is paused ?
- bool MPTK_IsPlaying `[get]`
  Is Midi file is playing ?
- TimeSpan MPTK_Duration `[get]`
  Value updated only when playing in Unity (for inspector refresh)
- TimeSpan MPTK_RealDuration `[get]`
  Real Duration of the midi calculated with all the midi Change Tempo Events find inside the midi file. Experimental!
- long MPTK_TickLast `[get]`
  Last tick position in Midi: Value of the tick for the last midi event in sequence expressed in number of "ticks". MPTK_TickLast / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.
- long MPTK_TickCurrent `[get, set]`
  Current tick position in Midi: Time of the current midi event expressed in number of "ticks". MPTK_TickCurrent / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.
- double MPTK_PulseLenght `[get]`
  Lenght in millisecond of a quarter
- TimeSpan MPTK_PlayTime `[get]`
  Updated only when playing in Unity (for inspector refresh)
- bool MPTK_LogEvents `[get, set]`
  Log midi events
- bool MPTK_KeepNoteOff `[get, set]`
  Should keep note off event Events from the Midi file ?
- int MPTK_Quantization `[get, set]`
  Level of quantization :
- List< TrackMidiEvent > MPTK_MidiEvents `[get]`
  [DEPRECATED] Get all the raw midi events available in the midi file. Use rather the class MidiLoad.
- int MPTK_DeltaTicksPerQuarterNote `[get]`
  Delta Ticks Per Quarter Note. Indicate the duration time in "ticks" which make up a quarter-note. For instance, if 96, then a duration of an eighth-note in the file would be 48.
- int MPTK_IndexSynthRate `[get, set]`
  Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.
- int MPTK_IndexSynthBuffSize `[get, set]`
  Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.
- float MPTK_MaxDistance `[get, set]`
  MaxDistance to use for PauseOnDistance
- float MPTK_Volume `[get, set]`
  Volume of midi playing. Must be >=0 and <= 1
- int MPTK_Transpose `[get, set]`
  Transpose note from -24 to 24

Detailed Description

[MPTK PRO] - Script for the prefab MidiExternalPlayer. See full example TestMidiExternalPlayer.cs with a light sequencer. Play a midi file from a path on the local deskop or from a web site

---

Member Function Documentation

new void MidiPlayerTK.MidiExternalPlayer.MPTK_Play ()

Play the midi file defined in MPTK_MidiName

```
 MidiExternalPlayer midiExternalPlayer = FindObjectOfType<MidiExternalPlayer>();
 MidiExternalPlayer.MPTK_MidiName = @"C:\Users\xxx\Midi\Bach The Art of Fugue - 
No1.mid";
     //or
 MidiExternalPlayer.MPTK_MidiName = 
"http://www.midiworld.com/midis/other/bach/bwv1060b.mid";
 MidiExternalPlayer.MPTK_Play();
!
```

new void MidiPlayerTK.MidiExternalPlayer.MPTK_Next ()

Play next Midi - NO EFFECT for external

new void MidiPlayerTK.MidiExternalPlayer.MPTK_Previous ()

Play previous Midi - NO EFFECT for external

void MidiPlayerTK.MidiFilePlayer.MPTK_Stop () [inherited]

Stop playing

void MidiPlayerTK.MidiFilePlayer.MPTK_RePlay () [inherited]

Restart playing of the current midi file

void MidiPlayerTK.MidiFilePlayer.MPTK_Pause (float *timeToPauseMS* = -1f) [inherited]

Pause the current playing

Parameters:

timeToPauseMS	time to pause in milliseconds. default: indefinitely

void MidiPlayerTK.MidiFilePlayer.MPTK_UnPause () [inherited]

Pause the current playing

Parameters:

timeToPauseMS	time to pause in milliseconds. default: indefinitely

MPTKEvent.EnumLength MidiPlayerTK.MidiFilePlayer.MPTK_NoteLength (MPTKEvent *note*) [inherited]

Return note length as https://en.wikipedia.org/wiki/Note_value

Parameters:

note	

Returns:

MPTKEvent.EnumLength

MidiLoad MidiPlayerTK.MidiFilePlayer.MPTK_Load () [inherited]

Load the midi file defined with MPTK_MidiName or MPTK_MidiIndex. It's an optional action before playing a midi file witk MPTK_Play.

```
 private void GetMidiInfo()
 {
    MidiLoad midiloaded = midiFilePlayer.MPTK_Load();
    if (midiloaded != null)
    {
       infoMidi = "Duration: " + midiloaded.MPTK Duration.TotalSeconds + "
seconds\n";
       infoMidi += "Tempo: " + midiloaded.MPTK InitialTempo + "\n";
       List<MPTKEvent> listEvents = midiloaded.MPTK_ReadMidiEvents();
       infoMidi += "Count Midi Events: " + listEvents.Count + "\n";
       Debug.Log(infoMidi);
    }
 }
!
```

Returns:

MidiLoad to access all the properties of the midi loaded

void MidiPlayerTK.MidiSynth.MPTK_InitSynth (int *channelCount* = 16) [inherited]

Init the synthetizer. Prefabs automatically initialize the synthetizer (see events). It's not usefull to call this method.

Parameters:

channelCount	Number of channel to create, default 16. Any other values are experimental!

void MidiPlayerTK.MidiSynth.MPTK_ClearAllSound (bool *destroyAudioSource* = false)[inherited]

Clear all sound

Parameters:

destroyAudioSource	Destroy also audioSource (default:false)

```
  if (GUILayout.Button("Clear"))
     midiStreamPlayer.MPTK ClearAllSound(true);
!
```

Member Data Documentation

EventNotesMidiClass MidiPlayerTK.MidiFilePlayer.OnEventNotesMidi[inherited]

Define unity event to trigger when notes available from the Midi file.

```
 MidiFilePlayer midiFilePlayer = FindObjectOfType<MidiFilePlayer>();
          ...
 if (!midiFilePlayer.OnEventNotesMidi.HasEvent())
 {
    // No listener defined, set now by script. NotesToPlay will be called for each
new notes read from Midi file
    midiFilePlayer.OnEventNotesMidi.AddListener(NotesToPlay);
 }
          ...
 public void NotesToPlay(List<MPTKEvent> notes)
 {
    Debug.Log(notes.Count);
    foreach (MPTKEvent midievent in notes)
    {
         ...
    }
 }
!
```

EventStartMidiClass MidiPlayerTK.MidiFilePlayer.OnEventStartPlayMidi[inherited]

Define unity event to trigger at start of playing the Midi.

```
! MidiFilePlayer midiFilePlayer = FindObjectOfType<MidiFilePlayer>();
          ...
 if (!midiFilePlayer.OnEventStartPlayMidi.HasEvent())
 {
    // No listener defined, set now by script. StartPlay will be called.
    midiFilePlayer.OnEventStartPlayMidi.AddListener(StartPlay);
 }
          ...
 public void StartPlay(string midiname)
 {
    Debug.LogFormat("Start playing midi {0}", midiname);
```

```
 }
!
```

EventEndMidiClass MidiPlayerTK.MidiFilePlayer.OnEventEndPlayMidi [inherited]

Define unity event to trigger at end of playing the midi.

```
 MidiFilePlayer midiFilePlayer = FindObjectOfType<MidiFilePlayer>();
         ...
 if (!midiFilePlayer.OnEventEndPlayMidi.HasEvent())
 {
    // No listener defined, set now by script. EndPlay will be called.
    midiFilePlayer.OnEventEndPlayMidi.AddListener(EndPlay);
 }
         ...
 public void EndPlay(string midiname, EventEndMidiEnum reason)
 {
    Debug.LogFormat("End playing midi {0} reason:{1}", midiname, reason);
 }
!
```

bool MidiPlayerTK.MidiSynth.MPTK_CorePlayer [inherited]

If true then Midi events are read and play from a dedicated thread. If false, MidiSynth will use AudioSource gameobject to play sound. This properties must be defined before running the application from the inspector. The default is true.

bool MidiPlayerTK.MidiSynth.MPTK_DirectSendToPlayer [inherited]

If true (default) then Midi events are sent automatically to the midi player. Set to false if you want to process events without playing sound. OnEventNotesMidi Unity Event can be used to process each notes.

bool MidiPlayerTK.MidiSynth.MPTK_EnableChangeTempo [inherited]

Should accept change tempo from Midi Events ?

bool MidiPlayerTK.MidiSynth.MPTK_PauseOnDistance [inherited]

Should the Midi playing must be paused if distance between AudioListener and MidiFilePlayer is greater than MaxDistance

bool MidiPlayerTK.MidiSynth.MPTK_EnablePanChange [inherited]

Should change pan from Midi Events or from SoundFont ?

bool MidiPlayerTK.MidiSynth.MPTK_EnablePresetDrum `[inherited]`

Should accept change Preset for Drum canal 10 ? Disabled by default. Could sometimes create bad sound with midi files not really compliant with the Midi norm.

bool MidiPlayerTK.MidiSynth.MPTK_LogWave `[inherited]`

Log for each wave to be played

uint MidiPlayerTK.MidiSynth.MPTK_ReleaseTimeMin = 500000 `[inherited]`

[Only when CorePlayer=False] Define a minimum release time at noteoff in 100 iem nanoseconds. Default 50 ms is a good tradeoff. Below some unpleasant sound could be heard. Useless when MPTK_CorePlayer is true.

int MidiPlayerTK.MidiSynth.MPTK_StatVoiceCountActive `[inherited]`

Count of the active voices (playing) - Readonly

int MidiPlayerTK.MidiSynth.MPTK_StatVoiceCountFree `[inherited]`

Count of the free voices for reusing on need. Older than AutoCleanVoiceTime are removed when count is over than AutoCleanVoiceLimit - Readonly

float MidiPlayerTK.MidiSynth.MPTK_StatVoiceRatioReused `[inherited]`

Percentage of voice reused during the synth life. 0: any reuse, 100:all voice reused (unattainable, of course!)

int MidiPlayerTK.MidiSynth.MPTK_StatVoicePlayed `[inherited]`

Count of voice played since the start of the synth

int MidiPlayerTK.MidiSynth.MPTK_AutoCleanVoiceLimit `[inherited]`

Free voices older than MPTK_AutoCleanVoiceLimit are removed when count is over than MPTK_AutoCleanVoiceTime

bool MidiPlayerTK.MidiSynth.MPTK_WeakDevice [inherited]

Should play on a weak device (cheaper smartphone) ? Apply only with AudioSource mode (MPTK_CorePlayer=False) Playing Midi files with WeakDevice activated could cause some bad interpretation of Midi Event, consequently bad sound.

EventSynthClass MidiPlayerTK.MidiSynth.OnEventSynthAwake [inherited]

Unity event fired at awake of the synthesizer. Name of the gameobject component is passed as a parameter.

```
 ...
 if (!midiStreamPlayer.OnEventSynthAwake.HasEvent())
    midiStreamPlayer.OnEventSynthAwake.AddListener(StartLoadingSynth);
 ...
 public void StartLoadingSynth(string name)
 {
     Debug.LogFormat("Synth {0} loading", name);
 }
!
```

EventSynthClass MidiPlayerTK.MidiSynth.OnEventSynthStarted [inherited]

Unity event fired at start of the synthesizer. Name of the gameobject component is passed as a parameter.

```
 ...
 if (!midiStreamPlayer.OnEventStartSynth.HasEvent())
    midiStreamPlayer.OnEventStartSynth.AddListener(EndLoadingSynth);
 ...
 public void EndLoadingSynth(string name)
 {
    Debug.LogFormat("Synth {0} loaded", name);
    midiStreamPlayer.MPTK_PlayEvent(
       new MPTKEvent() { Command = MPTKCommand.PatchChange, Value =
CurrentPatchInstrument, Channel = StreamChannel});
 }
!
```

Property Documentation

new string MidiPlayerTK.MidiExternalPlayer.MPTK_MidiName [get], [set]

Full path to Midi file or URL to play. must start with file:// or http:// or https://.

```
 MidiExternalPlayer midiExternalPlayer = FindObjectOfType<MidiExternalPlayer>();
 MidiExternalPlayer.MPTK_MidiName = @"C:\Users\xxx\Midi\Bach The Art of Fugue -
No1.mid";
     //or
 MidiExternalPlayer.MPTK_MidiName =
"http://www.midiworld.com/midis/other/bach/bwv1060b.mid";
 MidiExternalPlayer.MPTK_Play();
!
```

new int MidiPlayerTK.MidiExternalPlayer.MPTK_MidiIndex [get], [set]

Index Midi to play or playing - NO EFFECT for external

bool MidiPlayerTK.MidiFilePlayer.MPTK_PlayOnStart [get], [set], [inherited]

Should the Midi start playing when application starts ?

bool MidiPlayerTK.MidiFilePlayer.MPTK_Loop [get], [set], [inherited]

Should automatically restart when Midi reaches the end ?

double MidiPlayerTK.MidiFilePlayer.MPTK_Tempo [get], [inherited]

Get default tempo defined in Midi file or modified with Speed. Return QuarterPerMinuteValue similar to BPM (Beat Per Measure)

float MidiPlayerTK.MidiFilePlayer.MPTK_Speed [get], [set], [inherited]

Speed of playing. Between 0.1 (10%) to 5.0 (500%). Set to 1 for normal speed. Be carefull when modifying speed on fly from GUI. Each change generates 0.3s of pause, avoid little and frequent speed change.

double MidiPlayerTK.MidiFilePlayer.MPTK_Position [get], [set], [inherited]

Set or Get midi position time from 0 to lenght time of midi playing (in millisecond). No effect if the Midi is not playing.

```
 // Be carefull when modifying position on fly from GUI.
 // Each change generates 0.2s of pause, avoid little and frequent position
change.
 // Below change is applied only above 2 decimals.
 double currentPosition = Math.Round(midiFilePlayer.MPTK Position / 1000d, 2);
 double newPosition =
Math.Round(GUILayout.HorizontalSlider((float)currentPosition, 0f,
(float)midiFilePlayer.MPTK_Duration.TotalSeconds, GUILayout.Width(buttonWidth)),
2);
 if (newPosition != currentPosition)
 {
    Debug.Log("New position " + currentPosition + " --> " + newPosition );
    midiFilePlayer.MPTK_Position = newPosition * 1000d;
  }
!
```

bool MidiPlayerTK.MidiFilePlayer.MPTK_IsPaused [get], [inherited]

Is Midi file playing is paused ?

bool MidiPlayerTK.MidiFilePlayer.MPTK_IsPlaying [get], [inherited]

Is Midi file is playing ?

TimeSpan MidiPlayerTK.MidiFilePlayer.MPTK_Duration [get], [inherited]

Value updated only when playing in Unity (for inspector refresh)

Duration of the midi. This duration can change during the playing when Change Tempo Event are find inside the midi file.

TimeSpan MidiPlayerTK.MidiFilePlayer.MPTK_RealDuration [get], [inherited]

Real Duration of the midi calculated with all the midi Change Tempo Events find inside the midi file. Experimental!

long MidiPlayerTK.MidiFilePlayer.MPTK_TickLast [get], [inherited]

Last tick position in Midi: Value of the tick for the last midi event in sequence expressed in number of "ticks". MPTK_TickLast / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.

long MidiPlayerTK.MidiFilePlayer.MPTK_TickCurrent [get], [set], [inherited]

Current tick position in Midi: Time of the current midi event expressed in number of "ticks". MPTK_TickCurrent / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.

double MidiPlayerTK.MidiFilePlayer.MPTK_PulseLenght [get], [inherited]

Lenght in millisecond of a quarter

TimeSpan MidiPlayerTK.MidiFilePlayer.MPTK_PlayTime [get], [inherited]

Updated only when playing in Unity (for inspector refresh)

Time from the start of playing the current midi

bool MidiPlayerTK.MidiFilePlayer.MPTK_LogEvents [get], [set], [inherited]

Log midi events

bool MidiPlayerTK.MidiFilePlayer.MPTK_KeepNoteOff [get], [set], [inherited]

Should keep note off event Events from the Midi file ?

int MidiPlayerTK.MidiFilePlayer.MPTK_Quantization [get], [set], [inherited]

Level of quantization :

- 0 = None
- 1 = Quarter Note
- 2 = Eighth Note
- 3 = 16th Note
- 4 = 32th Note
- 5 = 64th Note

List<TrackMidiEvent> MidiPlayerTK.MidiFilePlayer.MPTK_MidiEvents [get], [inherited]

[DEPRECATED] Get all the raw midi events available in the midi file. Use rather the class MidiLoad.

```
 MidiLoad MidiLoaded = new MidiLoad();
 MidiLoaded.MPTK Load(midiindex);
 List<MPTKEvent> events = MidiLoaded.MPTK ReadMidiEvents();
!
```

int MidiPlayerTK.MidiFilePlayer.MPTK_DeltaTicksPerQuarterNote [get], [inherited]

Delta Ticks Per Quarter Note. Indicate the duration time in "ticks" which make up a quarter-note. For instance, if 96, then a duration of an eighth-note in the file would be 48.

int MidiPlayerTK.MidiSynth.MPTK_IndexSynthRate [get], [set], [inherited]

Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.

int MidiPlayerTK.MidiSynth.MPTK_IndexSynthBuffSize [get], [set], [inherited]

Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.

float MidiPlayerTK.MidiSynth.MPTK_MaxDistance `[get], [set], [inherited]`

MaxDistance to use for PauseOnDistance

float MidiPlayerTK.MidiSynth.MPTK_Volume `[get], [set], [inherited]`

Volume of midi playing. Must be >=0 and <= 1

int MidiPlayerTK.MidiSynth.MPTK_Transpose `[get], [set], [inherited]`

Transpose note from -24 to 24

MidiPlayerTK.MidiFileLoader

Script for the prefab MidiFilePlayer. Load a midi file. List of Midi file must be defined with Midi Player Setup (see Unity menu MPTK).
Inherits MonoBehaviour.

Public Member Functions

- void MPTK_Load (byte[] midiBytesToLoad=null)
  Load the midi file defined with MPTK_MidiName or MPTK_MidiIndex or from a array of bytes
- List< MPTKEvent > MPTK_ReadMidiEvents (long fromTicks=0, long toTicks=long.MaxValue)
  Read the list of midi events available in the Midi from a ticks position to an end position.
- void MPTK_Next ()
  Read next Midi from the list of midi defined in MPTK (see Unity menu Midi)
- void MPTK_Previous ()
  Read previous Midi from the list of midi defined in MPTK (see Unity menu Midi)
- MPTKEvent.EnumLength MPTK_NoteLength (MPTKEvent note)
  Return note length as https://en.wikipedia.org/wiki/Note_value

Properties

- string MPTK_MidiName `[get, set]`
  Midi name to load. Use the exact name defined in Unity resources folder MidiDB without any path or extension. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK.
- int MPTK_MidiIndex `[get, set]`
  Index Midi. Find the Index of Midi file from the popup in MidiFileLoader inspector. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK. return -1 if not found
- TimeSpan MPTK_Duration `[get]`
  Get duration of current Midi with current tempo
- long MPTK_TickLast `[get]`

Last tick position in Midi: Value of the tick for the last midi event in sequence expressed in number of "ticks". MPTK_TickLast / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.

- double MPTK_PulseLenght `[get]`
  Lenght in millisecond of a quarter
- bool MPTK_LogEvents `[get, set]`
  Updated only when playing in Unity (for inspector refresh)
- bool MPTK_KeepNoteOff `[get, set]`
  Should keep note off event Events ?
- int MPTK_Quantization `[get, set]`
  Level of quantization :
- int MPTK_DeltaTicksPerQuarterNote `[get]`
  Delta Ticks Per Quarter Note. Indicate the duration time in "ticks" which make up a quarter-note. For instance, if 96, then a duration of an eighth-note in the file would be 48.

Detailed Description

Script for the prefab MidiFilePlayer. Load a midi file. List of Midi file must be defined with Midi Player Setup (see Unity menu MPTK).

Member Function Documentation

void MidiPlayerTK.MidiFileLoader.MPTK_Load (byte [] *midiBytesToLoad* = `null`)

Load the midi file defined with MPTK_MidiName or MPTK_MidiIndex or from a array of bytes

Parameters:

midiBytesToLoad	

List<MPTKEvent> MidiPlayerTK.MidiFileLoader.MPTK_ReadMidiEvents (long *fromTicks* = `0`, long *toTicks* = `long.MaxValue`)

Read the list of midi events available in the Midi from a ticks position to an end position.

Parameters:

fromTicks	ticks start
toTicks	ticks end

Returns:

void MidiPlayerTK.MidiFileLoader.MPTK_Next ()

Read next Midi from the list of midi defined in MPTK (see Unity menu Midi)

void MidiPlayerTK.MidiFileLoader.MPTK_Previous ()

Read previous Midi from the list of midi defined in MPTK (see Unity menu Midi)

MPTKEvent.EnumLength MidiPlayerTK.MidiFileLoader.MPTK_NoteLength (MPTKEvent *note*)

Return note length as https://en.wikipedia.org/wiki/Note_value

Parameters:

note	

Returns:

MPTKEvent.EnumLength

Property Documentation

string MidiPlayerTK.MidiFileLoader.MPTK_MidiName [get], [set]

Midi name to load. Use the exact name defined in Unity resources folder MidiDB without any path or extension. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK.

```
 midiFileLoader.MPTK_MidiName = "Albinoni - Adagio";
!
```

int MidiPlayerTK.MidiFileLoader.MPTK_MidiIndex [get], [set]

Index Midi. Find the Index of Midi file from the popup in MidiFileLoader inspector. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK. return -1 if not found

Parameters:

index	

TimeSpan MidiPlayerTK.MidiFileLoader.MPTK_Duration [get]

Get duration of current Midi with current tempo

long MidiPlayerTK.MidiFileLoader.MPTK_TickLast [get]

Last tick position in Midi: Value of the tick for the last midi event in sequence expressed in number of "ticks". MPTK_TickLast / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.

double MidiPlayerTK.MidiFileLoader.MPTK_PulseLenght `[get]`

Lenght in millisecond of a quarter

bool MidiPlayerTK.MidiFileLoader.MPTK_LogEvents `[get]`, `[set]`

Updated only when playing in Unity (for inspector refresh)

Log midi events

bool MidiPlayerTK.MidiFileLoader.MPTK_KeepNoteOff `[get]`, `[set]`

Should keep note off event Events ?

int MidiPlayerTK.MidiFileLoader.MPTK_Quantization `[get]`, `[set]`

Level of quantization :

- `0` = None
- `1` = Quarter Note
- `2` = Eighth Note
- `3` = 16th Note
- `4` = 32th Note
- `5` = 64th Note

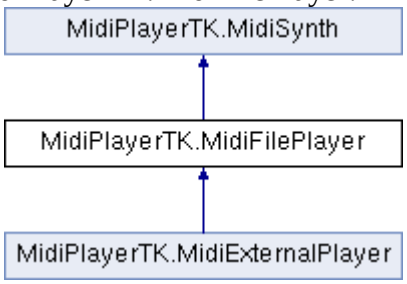
int MidiPlayerTK.MidiFileLoader.MPTK_DeltaTicksPerQuarterNote `[get]`

Delta Ticks Per Quarter Note. Indicate the duration time in "ticks" which make up a quarter-note. For instance, if 96, then a duration of an eighth-note in the file would be 48.

MidiPlayerTK.MidiFilePlayer

Play a midi file. Midi files must be defined with Midi Player Setup (see Unity menu MPTK) from Unity editor.

Inheritance diagram for MidiPlayerTK.MidiFilePlayer:



Public Member Functions

- void MPTK_Play ()
  Play the midi file defined with MPTK_MidiName or MPTK_MidiIndex
- void MPTK_Stop ()
  Stop playing
- void MPTK_RePlay ()
  Restart playing of the current midi file
- void MPTK_Pause (float timeToPauseMS=-1f)
  Pause the current playing
- void MPTK_UnPause ()
  Pause the current playing
- void MPTK_Next ()
  Play next Midi from the list of midi defined in MPTK (see Unity menu Midi)
- void MPTK_Previous ()
  Play previous Midi from the list of midi defined in MPTK (see Unity menu Midi)
- MPTKEvent.EnumLength MPTK_NoteLength (MPTKEvent note)
  Return note length as https://en.wikipedia.org/wiki/Note_value
- MidiLoad MPTK_Load ()
  Load the midi file defined with MPTK_MidiName or MPTK_MidiIndex. It's an optional action before playing a midi file witk MPTK_Play.
- void MPTK_InitSynth (int channelCount=16)
  Init the synthetizer. Prefabs automatically initialize the synthetizer (see events). It's not usefull to call this method.
- void MPTK_ClearAllSound (bool destroyAudioSource=false)
  Clear all sound

Public Attributes

- EventNotesMidiClass OnEventNotesMidi
  Define unity event to trigger when notes available from the Midi file.
- EventStartMidiClass OnEventStartPlayMidi
  Define unity event to trigger at start of playing the Midi.
- EventEndMidiClass OnEventEndPlayMidi
  Define unity event to trigger at end of playing the midi.
- bool MPTK_CorePlayer
  If true then Midi events are read and play from a dedicated thread. If false, MidiSynth will use AudioSource gameobject to play sound. This properties must be defined before running the application from the inspector. The default is true.
- bool MPTK_DirectSendToPlayer
  If true (default) then Midi events are sent automatically to the midi player. Set to false if you want to process events without playing sound. OnEventNotesMidi Unity Event can be used to process each notes.
- bool MPTK_EnableChangeTempo
  Should accept change tempo from Midi Events ?
- bool MPTK_PauseOnDistance
  Should the Midi playing must be paused if distance between AudioListener and MidiFilePlayer is greater than MaxDistance
- bool MPTK_EnablePanChange
  Should change pan from Midi Events or from SoundFont ?
- bool MPTK_EnablePresetDrum

Should accept change Preset for Drum canal 10 ? Disabled by default. Could sometimes create bad sound with midi files not really compliant with the Midi norm.

- bool MPTK_LogWave
  Log for each wave to be played
- uint MPTK_ReleaseTimeMin = 500000
  [Only when CorePlayer=False] Define a minimum release time at noteoff in 100 iem nanoseconds. Default 50 ms is a good tradeoff. Below some unpleasant sound could be heard. Useless when MPTK_CorePlayer is true.
- int MPTK_StatVoiceCountActive
  Count of the active voices (playing) - Readonly
- int MPTK_StatVoiceCountFree
  Count of the free voices for reusing on need. Older than AutoCleanVoiceTime are removed when count is over than AutoCleanVoiceLimit - Readonly
- float MPTK_StatVoiceRatioReused
  Percentage of voice reused during the synth life. 0: any reuse, 100:all voice reused (unattainable, of course!)
- int MPTK_StatVoicePlayed
  Count of voice played since the start of the synth
- int MPTK_AutoCleanVoiceLimit
  Free voices older than MPTK_AutoCleanVoiceLimit are removed when count is over than MPTK_AutoCleanVoiceTime
- bool MPTK_WeakDevice
  Should play on a weak device (cheaper smartphone) ? Apply only with AudioSource mode (MPTK_CorePlayer=False) Playing Midi files with WeakDevice activated could cause some bad interpretation of Midi Event, consequently bad sound.
- EventSynthClass OnEventSynthAwake
  Unity event fired at awake of the synthesizer. Name of the gameobject component is passed as a parameter.
- EventSynthClass OnEventSynthStarted
  Unity event fired at start of the synthesizer. Name of the gameobject component is passed as a parameter.

Properties

- string MPTK_MidiName `[get, set]`
  Midi name to play. Use the exact name defined in Unity resources folder MidiDB without any path or extension. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK.
- int MPTK_MidiIndex `[get, set]`
  Index Midi. Find the Index of Midi file (same values ad from the popup in MidiFilePlayer inspector). Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK. return -1 if not found
- bool MPTK_PlayOnStart `[get, set]`
  Should the Midi start playing when application starts ?
- bool MPTK_Loop `[get, set]`
  Should automatically restart when Midi reaches the end ?
- double MPTK_Tempo `[get]`
  Get default tempo defined in Midi file or modified with Speed. Return QuarterPerMinuteValue similar to BPM (Beat Per Measure)
- float MPTK_Speed `[get, set]`
  Speed of playing. Between 0.1 (10%) to 5.0 (500%). Set to 1 for normal speed. Be carefull when modifying speed on fly from GUI. Each change generates 0.3s of pause, avoid little and frequent speed change.

- double MPTK_Position `[get, set]`
  Set or Get midi position time from 0 to lenght time of midi playing (in millisecond). No effect if the Midi is not playing.
- bool MPTK_IsPaused `[get]`
  Is Midi file playing is paused ?
- bool MPTK_IsPlaying `[get]`
  Is Midi file is playing ?
- TimeSpan MPTK_Duration `[get]`
  Value updated only when playing in Unity (for inspector refresh)
- TimeSpan MPTK_RealDuration `[get]`
  Real Duration of the midi calculated with all the midi Change Tempo Events find inside the midi file. Experimental!
- long MPTK_TickLast `[get]`
  Last tick position in Midi: Value of the tick for the last midi event in sequence expressed in number of "ticks". MPTK_TickLast / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.
- long MPTK_TickCurrent `[get, set]`
  Current tick position in Midi: Time of the current midi event expressed in number of "ticks". MPTK_TickCurrent / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.
- double MPTK_PulseLenght `[get]`
  Lenght in millisecond of a quarter
- TimeSpan MPTK_PlayTime `[get]`
  Updated only when playing in Unity (for inspector refresh)
- bool MPTK_LogEvents `[get, set]`
  Log midi events
- bool MPTK_KeepNoteOff `[get, set]`
  Should keep note off event Events from the Midi file ?
- int MPTK_Quantization `[get, set]`
  Level of quantization :
- List< TrackMidiEvent > MPTK_MidiEvents `[get]`
  [DEPRECATED] Get all the raw midi events available in the midi file. Use rather the class MidiLoad.
- int MPTK_DeltaTicksPerQuarterNote `[get]`
  Delta Ticks Per Quarter Note. Indicate the duration time in "ticks" which make up a quarter-note. For instance, if 96, then a duration of an eighth-note in the file would be 48.
- int MPTK_IndexSynthRate `[get, set]`
  Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.
- int MPTK_IndexSynthBuffSize `[get, set]`
  Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.
- float MPTK_MaxDistance `[get, set]`
  MaxDistance to use for PauseOnDistance
- float MPTK_Volume `[get, set]`
  Volume of midi playing. Must be >=0 and <= 1
- int MPTK_Transpose `[get, set]`
  Transpose note from -24 to 24

Detailed Description

Play a midi file. Midi files must be defined with Midi Player Setup (see Unity menu MPTK) from Unity editor.

Member Function Documentation

void MidiPlayerTK.MidiFilePlayer.MPTK_Play ()

Play the midi file defined with MPTK_MidiName or MPTK_MidiIndex

void MidiPlayerTK.MidiFilePlayer.MPTK_Stop ()

Stop playing

void MidiPlayerTK.MidiFilePlayer.MPTK_RePlay ()

Restart playing of the current midi file

void MidiPlayerTK.MidiFilePlayer.MPTK_Pause (float *timeToPauseMS* = -1f)

Pause the current playing

Parameters:

timeToPauseMS	time to pause in milliseconds. default: indefinitely

void MidiPlayerTK.MidiFilePlayer.MPTK_UnPause ()

Pause the current playing

Parameters:

timeToPauseMS	time to pause in milliseconds. default: indefinitely

void MidiPlayerTK.MidiFilePlayer.MPTK_Next ()

Play next Midi from the list of midi defined in MPTK (see Unity menu Midi)

void MidiPlayerTK.MidiFilePlayer.MPTK_Previous ()

Play previous Midi from the list of midi defined in MPTK (see Unity menu Midi)

MPTKEvent.EnumLength MidiPlayerTK.MidiFilePlayer.MPTK_NoteLength (MPTKEvent *note*)

Return note length as https://en.wikipedia.org/wiki/Note_value

Parameters:

note	

Returns:

MPTKEvent.EnumLength

MidiLoad MidiPlayerTK.MidiFilePlayer.MPTK_Load ()

Load the midi file defined with MPTK_MidiName or MPTK_MidiIndex. It's an optional action before playing a midi file witk MPTK_Play.

```
 private void GetMidiInfo()
 {
    MidiLoad midiloaded = midiFilePlayer.MPTK_Load();
    if (midiloaded != null)
    {
       infoMidi = "Duration: " + midiloaded.MPTK Duration.TotalSeconds + "
seconds\n";
       infoMidi += "Tempo: " + midiloaded.MPTK InitialTempo + "\n";
       List<MPTKEvent> listEvents = midiloaded.MPTK_ReadMidiEvents();
       infoMidi += "Count Midi Events: " + listEvents.Count + "\n";
       Debug.Log(infoMidi);
    }
 }
!
```

Returns:

MidiLoad to access all the properties of the midi loaded

void MidiPlayerTK.MidiSynth.MPTK_InitSynth (int *channelCount* = 16) [inherited]

Init the synthetizer. Prefabs automatically initialize the synthetizer (see events). It's not usefull to call this method.

Parameters:

channelCount	Number of channel to create, default 16. Any other values are experimental!

void MidiPlayerTK.MidiSynth.MPTK_ClearAllSound (bool *destroyAudioSource* = false) [inherited]

Clear all sound

Parameters:

destroyAudioSource	Destroy also audioSource (default:false)

```
  if (GUILayout.Button("Clear"))
     midiStreamPlayer.MPTK_ClearAllSound(true);
!
```

Member Data Documentation

EventNotesMidiClass MidiPlayerTK.MidiFilePlayer.OnEventNotesMidi

Define unity event to trigger when notes available from the Midi file.

```
 MidiFilePlayer midiFilePlayer = FindObjectOfType<MidiFilePlayer>();
          ...
 if (!midiFilePlayer.OnEventNotesMidi.HasEvent())
 {
    // No listener defined, set now by script. NotesToPlay will be called for each
new notes read from Midi file
    midiFilePlayer.OnEventNotesMidi.AddListener(NotesToPlay);
 }
          ...
 public void NotesToPlay(List<MPTKEvent> notes)
 {
    Debug.Log(notes.Count);
    foreach (MPTKEvent midievent in notes)
    {
          ...
    }
 }
!
```

EventStartMidiClass MidiPlayerTK.MidiFilePlayer.OnEventStartPlayMidi

Define unity event to trigger at start of playing the Midi.

```
! MidiFilePlayer midiFilePlayer = FindObjectOfType<MidiFilePlayer>();
          ...
 if (!midiFilePlayer.OnEventStartPlayMidi.HasEvent())
 {
    // No listener defined, set now by script. StartPlay will be called.
    midiFilePlayer.OnEventStartPlayMidi.AddListener(StartPlay);
 }
          ...
 public void StartPlay(string midiname)
 {
    Debug.LogFormat("Start playing midi {0}", midiname);
 }
!
```

EventEndMidiClass MidiPlayerTK.MidiFilePlayer.OnEventEndPlayMidi

Define unity event to trigger at end of playing the midi.

```
 MidiFilePlayer midiFilePlayer = FindObjectOfType<MidiFilePlayer>();
          ...
 if (!midiFilePlayer.OnEventEndPlayMidi.HasEvent())
 {
    // No listener defined, set now by script. EndPlay will be called.
```

```
    midiFilePlayer.OnEventEndPlayMidi.AddListener(EndPlay);
 }
         ...
 public void EndPlay(string midiname, EventEndMidiEnum reason)
 {
    Debug.LogFormat("End playing midi {0} reason:{1}", midiname, reason);
 }
!
```

bool MidiPlayerTK.MidiSynth.MPTK_CorePlayer [inherited]

If true then Midi events are read and play from a dedicated thread. If false, MidiSynth will use AudioSource gameobject to play sound. This properties must be defined before running the application from the inspector. The default is true.

bool MidiPlayerTK.MidiSynth.MPTK_DirectSendToPlayer [inherited]

If true (default) then Midi events are sent automatically to the midi player. Set to false if you want to process events without playing sound. OnEventNotesMidi Unity Event can be used to process each notes.

bool MidiPlayerTK.MidiSynth.MPTK_EnableChangeTempo [inherited]

Should accept change tempo from Midi Events ?

bool MidiPlayerTK.MidiSynth.MPTK_PauseOnDistance [inherited]

Should the Midi playing must be paused if distance between AudioListener and MidiFilePlayer is greater than MaxDistance

bool MidiPlayerTK.MidiSynth.MPTK_EnablePanChange [inherited]

Should change pan from Midi Events or from SoundFont ?

bool MidiPlayerTK.MidiSynth.MPTK_EnablePresetDrum [inherited]

Should accept change Preset for Drum canal 10 ? Disabled by default. Could sometimes create bad sound with midi files not really compliant with the Midi norm.

bool MidiPlayerTK.MidiSynth.MPTK_LogWave [inherited]

Log for each wave to be played

uint MidiPlayerTK.MidiSynth.MPTK_ReleaseTimeMin = 500000 `[inherited]`

[Only when CorePlayer=False] Define a minimum release time at noteoff in 100 iem nanoseconds. Default 50 ms is a good tradeoff. Below some unpleasant sound could be heard. Useless when MPTK_CorePlayer is true.

int MidiPlayerTK.MidiSynth.MPTK_StatVoiceCountActive `[inherited]`

Count of the active voices (playing) - Readonly

int MidiPlayerTK.MidiSynth.MPTK_StatVoiceCountFree `[inherited]`

Count of the free voices for reusing on need. Older than AutoCleanVoiceTime are removed when count is over than AutoCleanVoiceLimit - Readonly

float MidiPlayerTK.MidiSynth.MPTK_StatVoiceRatioReused `[inherited]`

Percentage of voice reused during the synth life. 0: any reuse, 100:all voice reused (unattainable, of course!)

int MidiPlayerTK.MidiSynth.MPTK_StatVoicePlayed `[inherited]`

Count of voice played since the start of the synth

int MidiPlayerTK.MidiSynth.MPTK_AutoCleanVoiceLimit `[inherited]`

Free voices older than MPTK_AutoCleanVoiceLimit are removed when count is over than MPTK_AutoCleanVoiceTime

bool MidiPlayerTK.MidiSynth.MPTK_WeakDevice `[inherited]`

Should play on a weak device (cheaper smartphone) ? Apply only with AudioSource mode (MPTK_CorePlayer=False) Playing Midi files with WeakDevice activated could cause some bad interpretation of Midi Event, consequently bad sound.

EventSynthClass MidiPlayerTK.MidiSynth.OnEventSynthAwake `[inherited]`

Unity event fired at awake of the synthesizer. Name of the gameobject component is passed as a parameter.

```
 ...
 if (!midiStreamPlayer.OnEventSynthAwake.HasEvent())
    midiStreamPlayer.OnEventSynthAwake.AddListener(StartLoadingSynth);
 ...
 public void StartLoadingSynth(string name)
 {
     Debug.LogFormat("Synth {0} loading", name);
 }
!
```

EventSynthClass MidiPlayerTK.MidiSynth.OnEventSynthStarted `[inherited]`

Unity event fired at start of the synthesizer. Name of the gameobject component is passed as a parameter.

```
 ...
 if (!midiStreamPlayer.OnEventStartSynth.HasEvent())
    midiStreamPlayer.OnEventStartSynth.AddListener(EndLoadingSynth);
 ...
 public void EndLoadingSynth(string name)
 {
    Debug.LogFormat("Synth {0} loaded", name);
    midiStreamPlayer.MPTK PlayEvent(
       new MPTKEvent() { Command = MPTKCommand.PatchChange, Value =
CurrentPatchInstrument, Channel = StreamChannel});
 }
!
```

Property Documentation

string MidiPlayerTK.MidiFilePlayer.MPTK_MidiName `[get]`, `[set]`

Midi name to play. Use the exact name defined in Unity resources folder MidiDB without any path or extension. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK.

```
 midiFilePlayer.MPTK MidiName = "Albinoni - Adagio";
!
```

int MidiPlayerTK.MidiFilePlayer.MPTK_MidiIndex `[get]`, `[set]`

Index Midi. Find the Index of Midi file (same values ad from the popup in MidiFilePlayer inspector). Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK. return -1 if not found

Parameters:

index	

bool MidiPlayerTK.MidiFilePlayer.MPTK_PlayOnStart `[get]`, `[set]`

Should the Midi start playing when application starts ?

bool MidiPlayerTK.MidiFilePlayer.MPTK_Loop [get], [set]

Should automatically restart when Midi reaches the end ?

double MidiPlayerTK.MidiFilePlayer.MPTK_Tempo [get]

Get default tempo defined in Midi file or modified with Speed. Return QuarterPerMinuteValue similar to BPM (Beat Per Measure)

float MidiPlayerTK.MidiFilePlayer.MPTK_Speed [get], [set]

Speed of playing. Between 0.1 (10%) to 5.0 (500%). Set to 1 for normal speed. Be carefull when modifying speed on fly from GUI. Each change generates 0.3s of pause, avoid little and frequent speed change.

double MidiPlayerTK.MidiFilePlayer.MPTK_Position [get], [set]

Set or Get midi position time from 0 to lenght time of midi playing (in millisecond). No effect if the Midi is not playing.

```
 // Be carefull when modifying position on fly from GUI.
 // Each change generates 0.2s of pause, avoid little and frequent position
change.
 // Below change is applied only above 2 decimals.
 double currentPosition = Math.Round(midiFilePlayer.MPTK_Position / 1000d, 2);
 double newPosition =
Math.Round(GUILayout.HorizontalSlider((float)currentPosition, 0f,
(float)midiFilePlayer.MPTK Duration.TotalSeconds, GUILayout.Width(buttonWidth)),
2);
 if (newPosition != currentPosition)
 {
    Debug.Log("New position " + currentPosition + " --> " + newPosition );
    midiFilePlayer.MPTK Position = newPosition * 1000d;
  }
!
```

bool MidiPlayerTK.MidiFilePlayer.MPTK_IsPaused [get]

Is Midi file playing is paused ?

bool MidiPlayerTK.MidiFilePlayer.MPTK_IsPlaying [get]

Is Midi file is playing ?

TimeSpan MidiPlayerTK.MidiFilePlayer.MPTK_Duration [get]

Value updated only when playing in Unity (for inspector refresh)

Duration of the midi. This duration can change during the playing when Change Tempo Event are find inside the midi file.

TimeSpan MidiPlayerTK.MidiFilePlayer.MPTK_RealDuration [get]

Real Duration of the midi calculated with all the midi Change Tempo Events find inside the midi file. Experimental!

long MidiPlayerTK.MidiFilePlayer.MPTK_TickLast [get]

Last tick position in Midi: Value of the tick for the last midi event in sequence expressed in number of "ticks". MPTK_TickLast / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.

long MidiPlayerTK.MidiFilePlayer.MPTK_TickCurrent [get], [set]

Current tick position in Midi: Time of the current midi event expressed in number of "ticks". MPTK_TickCurrent / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.

double MidiPlayerTK.MidiFilePlayer.MPTK_PulseLenght [get]

Lenght in millisecond of a quarter

TimeSpan MidiPlayerTK.MidiFilePlayer.MPTK_PlayTime [get]

Updated only when playing in Unity (for inspector refresh)

Time from the start of playing the current midi

bool MidiPlayerTK.MidiFilePlayer.MPTK_LogEvents [get], [set]

Log midi events

bool MidiPlayerTK.MidiFilePlayer.MPTK_KeepNoteOff [get], [set]

Should keep note off event Events from the Midi file ?

int MidiPlayerTK.MidiFilePlayer.MPTK_Quantization [get], [set]

Level of quantization :

- 0 = None
- 1 = Quarter Note
- 2 = Eighth Note
- 3 = 16th Note
- 4 = 32th Note
- 5 = 64th Note

List<TrackMidiEvent> MidiPlayerTK.MidiFilePlayer.MPTK_MidiEvents [get]

[DEPRECATED] Get all the raw midi events available in the midi file. Use rather the class MidiLoad.

```
MidiLoad MidiLoaded = new MidiLoad();
MidiLoaded.MPTK_Load(midiindex);
List<MPTKEvent> events = MidiLoaded.MPTK_ReadMidiEvents();
!
```

int MidiPlayerTK.MidiFilePlayer.MPTK_DeltaTicksPerQuarterNote [get]

Delta Ticks Per Quarter Note. Indicate the duration time in "ticks" which make up a quarter-note. For instance, if 96, then a duration of an eighth-note in the file would be 48.

int MidiPlayerTK.MidiSynth.MPTK_IndexSynthRate [get], [set], [inherited]

Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.

int MidiPlayerTK.MidiSynth.MPTK_IndexSynthBuffSize [get], [set], [inherited]

Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.

float MidiPlayerTK.MidiSynth.MPTK_MaxDistance [get], [set], [inherited]

MaxDistance to use for PauseOnDistance

float MidiPlayerTK.MidiSynth.MPTK_Volume [get], [set], [inherited]

Volume of midi playing. Must be >=0 and <= 1

int MidiPlayerTK.MidiSynth.MPTK_Transpose [get], [set], [inherited]

Transpose note from -24 to 24

MidiPlayerTK.MidiFileWriter

[MPTK PRO] - Write a midi file from differents sources based on NAudio frawemork. See full example TestMidiWriter.cs with a light sequencer.

Public Member Functions

- MidiFileWriter ()
  Create an empty MidiFileWriter
- MidiFileWriter (int deltaTicksPerQuarterNote, int midiFileType)
  Create a MidiFileWriter with an empty Midi Event list
- bool MPTK_LoadFromMPTK (List< TrackMidiEvent > MidiSorted)
  Create a MidiFileWriter from a MPTK list of midi events. A midi file must be loaded before from a MidiFilePlayer gameobject (as in example) or from a call to MidiFileWriter.MPTK_LoadFromFile(filename).
- bool MPTK_LoadFromMidiDB (int indexMidiDb)
  Create a MidiFileWriter from a Midi found in MPTK MidiDB
- void MPTK_CreateTrack (int count)
  Create tracks
- void MPTK_EndTrack (int trackNumber)
  Close the track (mandatory for a well formed midi file)
- void MPTK_AddEvent (int track, MidiEvent midievent)
  Add a generic Midi event
- void MPTK_AddNote (int track, long absoluteTime, int channel, int note, int velocity, int duration)
  Add a note event. the corresponding Noteoff is automatically created.
- bool MPTK_LoadFromFile (string filename)
  Load a Midi file from OS system file (could be dependant of the OS)
- bool MPTK_WriteToFile (string filename)
  Write Midi file to an OS folder
- bool MPTK_WriteToMidiDB (string filename)
  Write Midi file to MidiDB. To be used only in edit mode not in a standalone application.

Static Public Member Functions

- static int MPTK_GetMicrosecondsPerQuaterNote (int bpm)
  Convert BPM to duration or a quarter in microsecond

Properties

- int MPTK_DeltaTicksPerQuarterNote `[get]`
  Get the DeltaTicksPerQuarterNote of the loaded midi
- int MPTK_TrackCount `[get]`
  Get the track count of the loaded midi
- int MPTK_MidiFileType `[get]`
  Get the midi file type of the loaded midi (0,1,2)

Detailed Description

[MPTK PRO] - Write a midi file from differents sources based on NAudio frawemork. See full example TestMidiWriter.cs with a light sequencer.

Constructor & Destructor Documentation

MidiPlayerTK.MidiFileWriter.MidiFileWriter ()

Create an empty MidiFileWriter

MidiPlayerTK.MidiFileWriter.MidiFileWriter (int *deltaTicksPerQuarterNote*, int *midiFileType*)

Create a MidiFileWriter with an empty Midi Event list

Parameters:

deltaTicksPerQuarterNote	
midiFileType	

Member Function Documentation

bool MidiPlayerTK.MidiFileWriter.MPTK_LoadFromMPTK (List< TrackMidiEvent > *MidiSorted*)

Create a MidiFileWriter from a MPTK list of midi events. A midi file must be loaded before from a MidiFilePlayer gameobject (as in example) or from a call to MidiFileWriter.MPTK_LoadFromFile(filename).

Parameters:

MidiSorted	

bool MidiPlayerTK.MidiFileWriter.MPTK_LoadFromMidiDB (int *indexMidiDb*)

Create a MidiFileWriter from a Midi found in MPTK MidiDB

Parameters:

indexMidiDb	

void MidiPlayerTK.MidiFileWriter.MPTK_CreateTrack (int *count*)

Create tracks

Parameters:

count	number of tracks to create

void MidiPlayerTK.MidiFileWriter.MPTK_EndTrack (int *trackNumber*)

Close the track (mandatory for a well formed midi file)

Parameters:

trackNumber	Track number to close

void MidiPlayerTK.MidiFileWriter.MPTK_AddEvent (int *track*, MidiEvent *midievent*)

Add a generic Midi event

Parameters:

track	
midievent	

void MidiPlayerTK.MidiFileWriter.MPTK_AddNote (int *track*, long *absoluteTime*, int *channel*, int *note*, int *velocity*, int *duration*)

Add a note event. the corresponding Noteoff is automatically created.

Parameters:

track	
absoluteTime	
channel	
note	
velocity	
duration	

static int MidiPlayerTK.MidiFileWriter.MPTK_GetMicrosecondsPerQuaterNote (int *bpm*) `[static]`

Convert BPM to duration or a quarter in microsecond

Parameters:

bpm	beat per measure

Returns:

bool MidiPlayerTK.MidiFileWriter.MPTK_LoadFromFile (string *filename*)

Load a Midi file from OS system file (could be dependant of the OS)

Parameters:

filename	

Returns:

bool MidiPlayerTK.MidiFileWriter.MPTK_WriteToFile (string *filename*)

Write Midi file to an OS folder

Parameters:

filename	filename of the midi file

Returns:

bool MidiPlayerTK.MidiFileWriter.MPTK_WriteToMidiDB (string *filename*)

Write Midi file to MidiDB. To be used only in edit mode not in a standalone application.

Parameters:

filename	filename of the midi file without any folder and any extension

Returns:

Property Documentation

int MidiPlayerTK.MidiFileWriter.MPTK_DeltaTicksPerQuarterNote `[get]`

Get the DeltaTicksPerQuarterNote of the loaded midi

int MidiPlayerTK.MidiFileWriter.MPTK_TrackCount `[get]`

Get the track count of the loaded midi

int MidiPlayerTK.MidiFileWriter.MPTK_MidiFileType `[get]`

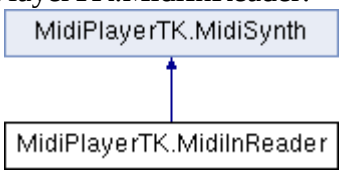
Get the midi file type of the loaded midi (0,1,2)

MidiPlayerTK.MidiInReader

Play generated notes. Any Midi file is necessary rather create music from your own algorithm with

MPTK_PlayEvent(). Duration can be set in the MPTKEvent, but a note can also be stopped with MPTK_StopEvent().

Inheritance diagram for MidiPlayerTK.MidiInReader:



Public Member Functions

- void MPTK_InitSynth (int channelCount=16)
  Init the synthetizer. Prefabs automatically initialize the synthetizer (see events). It's not usefull to call this method.
- void MPTK_ClearAllSound (bool destroyAudioSource=false)
  Clear all sound

Public Attributes

- bool MPTK_ReadMidiInput
  Read Midi input
- bool MPTK_LogEvents
  Log midi events
- EventMidiClass OnEventInputMidi
  Define unity event to trigger when note available from the Midi file.
- bool MPTK_CorePlayer
  If true then Midi events are read and play from a dedicated thread. If false, MidiSynth will use AudioSource gameobject to play sound. This properties must be defined before running the application from the inspector. The default is true.
- bool MPTK_DirectSendToPlayer
  If true (default) then Midi events are sent automatically to the midi player. Set to false if you want to process events without playing sound. OnEventNotesMidi Unity Event can be used to process each notes.
- bool MPTK_EnableChangeTempo
  Should accept change tempo from Midi Events ?
- bool MPTK_PauseOnDistance
  Should the Midi playing must be paused if distance between AudioListener and MidiFilePlayer is greater than MaxDistance
- bool MPTK_EnablePanChange
  Should change pan from Midi Events or from SoundFont ?
- bool MPTK_EnablePresetDrum
  Should accept change Preset for Drum canal 10 ? Disabled by default. Could sometimes create bad sound with midi files not really compliant with the Midi norm.
- bool MPTK_LogWave
  Log for each wave to be played
- uint MPTK_ReleaseTimeMin = 500000
  [Only when CorePlayer=False] Define a minimum release time at noteoff in 100 iem nanoseconds. Default 50 ms is a good tradeoff. Below some unpleasant sound could be heard. Useless when MPTK_CorePlayer is true.
- int MPTK_StatVoiceCountActive
  Count of the active voices (playing) - Readonly
- int MPTK_StatVoiceCountFree
  Count of the free voices for reusing on need. Older than AutoCleanVoiceTime are removed when count is over than AutoCleanVoiceLimit - Readonly

- float MPTK_StatVoiceRatioReused
  Percentage of voice reused during the synth life. 0: any reuse, 100:all voice reused (unattainable, of course!)
- int MPTK_StatVoicePlayed
  Count of voice played since the start of the synth
- int MPTK_AutoCleanVoiceLimit
  Free voices older than MPTK_AutoCleanVoiceLimit are removed when count is over than MPTK_AutoCleanVoiceTime
- bool MPTK_WeakDevice
  Should play on a weak device (cheaper smartphone) ? Apply only with AudioSource mode (MPTK_CorePlayer=False) Playing Midi files with WeakDevice activated could cause some bad interpretation of Midi Event, consequently bad sound.
- EventSynthClass OnEventSynthAwake
  Unity event fired at awake of the synthesizer. Name of the gameobject component is passed as a parameter.
- EventSynthClass OnEventSynthStarted
  Unity event fired at start of the synthesizer. Name of the gameobject component is passed as a parameter.

Properties

- int MPTK_IndexSynthRate `[get, set]`
  Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.
- int MPTK_IndexSynthBuffSize `[get, set]`
  Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.
- float MPTK_MaxDistance `[get, set]`
  MaxDistance to use for PauseOnDistance
- float MPTK_Volume `[get, set]`
  Volume of midi playing. Must be >=0 and <= 1
- int MPTK_Transpose `[get, set]`
  Transpose note from -24 to 24

Detailed Description

Play generated notes. Any Midi file is necessary rather create music from your own algorithm with MPTK_PlayEvent(). Duration can be set in the MPTKEvent, but a note can also be stopped with MPTK_StopEvent().

Member Function Documentation

void MidiPlayerTK.MidiSynth.MPTK_InitSynth (int *channelCount* = 16) `[inherited]`

Init the synthetizer. Prefabs automatically initialize the synthetizer (see events). It's not usefull to call this method.

Parameters:

channelCount	Number of channel to create, default 16. Any other values are experimental!

void MidiPlayerTK.MidiSynth.MPTK_ClearAllSound (bool *destroyAudioSource* = false)[inherited]

Clear all sound

Parameters:

destroyAudioSource	Destroy also audioSource (default:false)

```
  if (GUILayout.Button("Clear"))
     midiStreamPlayer.MPTK ClearAllSound(true);
!
```

Member Data Documentation

bool MidiPlayerTK.MidiInReader.MPTK_ReadMidiInput

Read Midi input

bool MidiPlayerTK.MidiInReader.MPTK_LogEvents

Log midi events

EventMidiClass MidiPlayerTK.MidiInReader.OnEventInputMidi

Define unity event to trigger when note available from the Midi file.

```
 MidiInReader midiFilePlayer = FindObjectOfType<MidiInReader>();
          ...
 if (!midiFilePlayer.OnEventInputMidi.HasEvent())
 {
    // No listener defined, set now by script. NotesToPlay will be called for each
new notes read from Midi file
    midiFilePlayer.OnEventInputMidi.AddListener(NotesToPlay);
 }
          ...
 public void NotesToPlay(MPTKEvent notes)
 {
    Debug.Log(notes.Value);
    foreach (MPTKEvent midievent in notes)
    {
          ...
    }
 }
!
```

bool MidiPlayerTK.MidiSynth.MPTK_CorePlayer [inherited]

If true then Midi events are read and play from a dedicated thread. If false, MidiSynth will use AudioSource gameobject to play sound. This properties must be defined before running the application from the inspector. The default is true.

bool MidiPlayerTK.MidiSynth.MPTK_DirectSendToPlayer [inherited]

If true (default) then Midi events are sent automatically to the midi player. Set to false if you want to process events without playing sound. OnEventNotesMidi Unity Event can be used to process each notes.

bool MidiPlayerTK.MidiSynth.MPTK_EnableChangeTempo [inherited]

Should accept change tempo from Midi Events ?

bool MidiPlayerTK.MidiSynth.MPTK_PauseOnDistance [inherited]

Should the Midi playing must be paused if distance between AudioListener and MidiFilePlayer is greater than MaxDistance

bool MidiPlayerTK.MidiSynth.MPTK_EnablePanChange [inherited]

Should change pan from Midi Events or from SoundFont ?

bool MidiPlayerTK.MidiSynth.MPTK_EnablePresetDrum [inherited]

Should accept change Preset for Drum canal 10 ? Disabled by default. Could sometimes create bad sound with midi files not really compliant with the Midi norm.

bool MidiPlayerTK.MidiSynth.MPTK_LogWave [inherited]

Log for each wave to be played

uint MidiPlayerTK.MidiSynth.MPTK_ReleaseTimeMin = 500000 [inherited]

[Only when CorePlayer=False] Define a minimum release time at noteoff in 100 iem nanoseconds. Default 50 ms is a good tradeoff. Below some unpleasant sound could be heard. Useless when MPTK_CorePlayer is true.

int MidiPlayerTK.MidiSynth.MPTK_StatVoiceCountActive [inherited]

Count of the active voices (playing) - Readonly

int MidiPlayerTK.MidiSynth.MPTK_StatVoiceCountFree [inherited]

Count of the free voices for reusing on need. Older than AutoCleanVoiceTime are removed when count is over than AutoCleanVoiceLimit - Readonly

float MidiPlayerTK.MidiSynth.MPTK_StatVoiceRatioReused [inherited]

Percentage of voice reused during the synth life. 0: any reuse, 100:all voice reused (unattainable, of course!)

int MidiPlayerTK.MidiSynth.MPTK_StatVoicePlayed [inherited]

Count of voice played since the start of the synth

int MidiPlayerTK.MidiSynth.MPTK_AutoCleanVoiceLimit [inherited]

Free voices older than MPTK_AutoCleanVoiceLimit are removed when count is over than MPTK_AutoCleanVoiceTime

bool MidiPlayerTK.MidiSynth.MPTK_WeakDevice [inherited]

Should play on a weak device (cheaper smartphone) ? Apply only with AudioSource mode (MPTK_CorePlayer=False) Playing Midi files with WeakDevice activated could cause some bad interpretation of Midi Event, consequently bad sound.

EventSynthClass MidiPlayerTK.MidiSynth.OnEventSynthAwake [inherited]

Unity event fired at awake of the synthesizer. Name of the gameobject component is passed as a parameter.

```
 ...
 if (!midiStreamPlayer.OnEventSynthAwake.HasEvent())
    midiStreamPlayer.OnEventSynthAwake.AddListener(StartLoadingSynth);
 ...
 public void StartLoadingSynth(string name)
 {
     Debug.LogFormat("Synth {0} loading", name);
 }
!
```

EventSynthClass MidiPlayerTK.MidiSynth.OnEventSynthStarted [inherited]

Unity event fired at start of the synthesizer. Name of the gameobject component is passed as a parameter.

```
 ...
 if (!midiStreamPlayer.OnEventStartSynth.HasEvent())
    midiStreamPlayer.OnEventStartSynth.AddListener(EndLoadingSynth);
 ...
 public void EndLoadingSynth(string name)
 {
    Debug.LogFormat("Synth {0} loaded", name);
    midiStreamPlayer.MPTK_PlayEvent(
       new MPTKEvent() { Command = MPTKCommand.PatchChange, Value =
CurrentPatchInstrument, Channel = StreamChannel});
 }
!
```

Property Documentation

int MidiPlayerTK.MidiSynth.MPTK_IndexSynthRate [get], [set], [inherited]

Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.

int MidiPlayerTK.MidiSynth.MPTK_IndexSynthBuffSize [get], [set], [inherited]

Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.

float MidiPlayerTK.MidiSynth.MPTK_MaxDistance [get], [set], [inherited]

MaxDistance to use for PauseOnDistance

float MidiPlayerTK.MidiSynth.MPTK_Volume [get], [set], [inherited]

Volume of midi playing. Must be >=0 and <= 1

int MidiPlayerTK.MidiSynth.MPTK_Transpose [get], [set], [inherited]

Transpose note from -24 to 24

MidiPlayerTK.MidiListPlayer

[MPTK PRO] - Script for the prefab MidiListPlayer. Play a list of pre-selected midi file from the dedicated inspector. List of Midi files must exists in MidiDB. See Midi Player Setup (Unity menu MPTK).
Inherits MonoBehaviour.

Classes

- class MPTK_MidiPlayItem
  Define a midi to be added in the list

Public Member Functions

- void MPTK_NewList ()
  Create an empty list
- void MPTK_AddMidi (string name, float start=0, float end=0)
  Add a Midi name to the list. Use the exact name defined in Unity resources (folder MidiDB) without any path or extension. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK.
- void MPTK_RemoveMidi (string name)
  Remove a Midi name from the list. Use the exact name defined in Unity resources folder MidiDB without any path or extension.
- void MPTK_RemoveMidiAt (int index)
  Remove a Midi at position from the list..
- MPTK_MidiPlayItem MPTK_GetAt (int index)
  Get description of a play item at position.
- void MPTK_ReIndexMidi ()
  Recalculate the index of the midi from the list.
- void MPTK_Play ()
  Play the midi in list at MPTK_PlayIndex position
- void MPTK_Stop ()
  Stop playing
- void MPTK_RePlay ()
  Restart playing the current midi file
- void MPTK_Pause (float timeToPauseMS=-1f)
  Pause the current playing
- void MPTK_UnPause ()
  Pause the current playing
- void MPTK_Next ()
  Play next Midi in list
- void MPTK_Previous ()
  Play previous Midi in list

Public Attributes

- List< MPTK_MidiPlayItem > MPTK_PlayList
  Play list
- EventStartMidiClass OnEventStartPlayMidi
  Define unity event to trigger at start
- EventEndMidiClass OnEventEndPlayMidi
  Define unity event to trigger at end
- MidiListPlayerStatus MPTK_MidiFilePlayer_1

First MidiFilePlayer to play the Midi

- MidiListPlayerStatus MPTK_MidiFilePlayer_2
  Second MidiFilePlayer to play the Midi
- float MPTK_OverlayTimeMS
  Duration of overlay between playing two midi

Properties

- float MPTK_Volume `[get, set]`
  Volume of midi playing. Must be >=0 and <= 1
- int MPTK_PlayIndex `[get, set]`
  Play a specific Midi in the list.
- bool MPTK_PlayOnStart `[get, set]`
  Should the Midi start playing when application start ?
- bool MPTK_Loop `[get, set]`
  Should automatically restart when Midi reach the end ?
- double MPTK_Position `[get, set]`
  Set or Get midi position time from 0 to lenght time of midi playing (in millisecond). No effect if the Midi is not playing.
- long MPTK_TickLast `[get]`
  Last tick position in Midi: Value of the tick for the last midi event in sequence expressed in number of "ticks". MPTK_TickLast / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.
- long MPTK_TickCurrent `[get, set]`
  Current tick position in Midi: Time of the current midi event expressed in number of "ticks". MPTK_TickCurrent / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.
- TimeSpan MPTK_Duration `[get]`
  Duration of the midi. This duration can change during the playing when Change Tempo Event are processed.
- TimeSpan MPTK_RealDuration `[get]`
  Real Duration of the midi calculated with all the midi Change Tempo Events find inside the midi file. Experimental!
- bool MPTK_IsPaused `[get]`
  Is Midi file playing is paused ?
- bool MPTK_IsPlaying `[get]`
  Is Midi file is playing ?

Detailed Description

[MPTK PRO] - Script for the prefab MidiListPlayer. Play a list of pre-selected midi file from the dedicated inspector. List of Midi files must exists in MidiDB. See Midi Player Setup (Unity menu MPTK).

Member Function Documentation

void MidiPlayerTK.MidiListPlayer.MPTK_NewList ()

Create an empty list

void MidiPlayerTK.MidiListPlayer.MPTK_AddMidi (string *name*, float *start* = 0, float *end* = 0)

Add a Midi name to the list. Use the exact name defined in Unity resources (folder MidiDB) without any path or extension. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK.

```
 midiListPlayer.MPTK_AddMidi("Albinoni - Adagio");
 midiListPlayer.MPTK_AddMidi("Conan The Barbarian", 10000, 20000);
!
```

Parameters:

name	midi filename as defined in resources
start	starting time of playing (ms). Default: start of the midi
end	endding time of playing (ms). Default: end of midi

void MidiPlayerTK.MidiListPlayer.MPTK_RemoveMidi (string *name*)

Remove a Midi name from the list. Use the exact name defined in Unity resources folder MidiDB without any path or extension.

```
// midiListPlayer.MPTK_RemoveMidi("Albinoni - Adagio");
```

void MidiPlayerTK.MidiListPlayer.MPTK_RemoveMidiAt (int *index*)

Remove a Midi at position from the list..

```
// midiListPlayer.MPTK_RemoveMidiAt(1);
```

MPTK_MidiPlayItem MidiPlayerTK.MidiListPlayer.MPTK_GetAt (int *index*)

Get description of a play item at position.

```
// midiListPlayer.MPTK_GetAt(1);
```

void MidiPlayerTK.MidiListPlayer.MPTK_ReIndexMidi ()

Recalculate the index of the midi from the list.

void MidiPlayerTK.MidiListPlayer.MPTK_Play ()

Play the midi in list at MPTK_PlayIndex position

void MidiPlayerTK.MidiListPlayer.MPTK_Stop ()

Stop playing

void MidiPlayerTK.MidiListPlayer.MPTK_RePlay ()

Restart playing the current midi file

void MidiPlayerTK.MidiListPlayer.MPTK_Pause (float *timeToPauseMS* = `-1f`)

Pause the current playing

Parameters:

timeToPauseMS	time to pause in milliseconds. default: indefinitely

void MidiPlayerTK.MidiListPlayer.MPTK_UnPause ()

Pause the current playing

Parameters:

timeToPauseMS	time to pause in milliseconds. default: indefinitely

void MidiPlayerTK.MidiListPlayer.MPTK_Next ()

Play next Midi in list

void MidiPlayerTK.MidiListPlayer.MPTK_Previous ()

Play previous Midi in list

Member Data Documentation

List<MPTK_MidiPlayItem> MidiPlayerTK.MidiListPlayer.MPTK_PlayList

Play list

EventStartMidiClass MidiPlayerTK.MidiListPlayer.OnEventStartPlayMidi

Define unity event to trigger at start

EventEndMidiClass MidiPlayerTK.MidiListPlayer.OnEventEndPlayMidi

Define unity event to trigger at end

MidiListPlayerStatus MidiPlayerTK.MidiListPlayer.MPTK_MidiFilePlayer_2

Second MidiFilePlayer to play the Midi

float MidiPlayerTK.MidiListPlayer.MPTK_OverlayTimeMS

Duration of overlay between playing two midi

Property Documentation

float MidiPlayerTK.MidiListPlayer.MPTK_Volume `[get], [set]`

Volume of midi playing. Must be >=0 and <= 1

int MidiPlayerTK.MidiListPlayer.MPTK_PlayIndex `[get], [set]`

Play a specific Midi in the list.

bool MidiPlayerTK.MidiListPlayer.MPTK_PlayOnStart `[get], [set]`

Should the Midi start playing when application start ?

bool MidiPlayerTK.MidiListPlayer.MPTK_Loop `[get], [set]`

Should automatically restart when Midi reach the end ?

double MidiPlayerTK.MidiListPlayer.MPTK_Position `[get], [set]`

Set or Get midi position time from 0 to lenght time of midi playing (in millisecond). No effect if the Midi is not playing.

```
 // Be carefull when modifying position on fly from GUI.
 // Each change generates 0.2s of pause, avoid little and frequent position
change.
 // Below change is applied only above 2 decimals.
 double currentPosition = Math.Round(midiFilePlayer.MPTK_Position / 1000d, 2);
 double newPosition =
Math.Round(GUILayout.HorizontalSlider((float)currentPosition, 0f,
(float)midiFilePlayer.MPTK RealDuration.TotalSeconds,
GUILayout.Width(buttonWidth)), 2);
 if (newPosition != currentPosition)
 {
    Debug.Log("New position " + currentPosition + " --> " + newPosition );
    midiFilePlayer.MPTK_Position = newPosition * 1000d;
  }
!
```

long MidiPlayerTK.MidiListPlayer.MPTK_TickLast `[get]`

Last tick position in Midi: Value of the tick for the last midi event in sequence expressed in number of "ticks". MPTK_TickLast / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.

long MidiPlayerTK.MidiListPlayer.MPTK_TickCurrent `[get]`, `[set]`

Current tick position in Midi: Time of the current midi event expressed in number of "ticks". MPTK_TickCurrent / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.

TimeSpan MidiPlayerTK.MidiListPlayer.MPTK_Duration `[get]`

Duration of the midi. This duration can change during the playing when Change Tempo Event are processed.

TimeSpan MidiPlayerTK.MidiListPlayer.MPTK_RealDuration `[get]`

Real Duration of the midi calculated with all the midi Change Tempo Events find inside the midi file. Experimental!

bool MidiPlayerTK.MidiListPlayer.MPTK_IsPaused `[get]`

Is Midi file playing is paused ?

bool MidiPlayerTK.MidiListPlayer.MPTK_IsPlaying `[get]`

Is Midi file is playing ?

MidiPlayerTK.MidiLoad

Base class for loading a Midi file. No seqencer, no synthetizer. Usefull to load all the Midi events from a Midi.

Public Member Functions

- bool MPTK_Load (int index, bool strict=false)
  Load Midi from midi MPTK referential (Unity resource). The index of the Midi file can be found in the windo "Midi File Setup". Display with menu MPTK / Midi File Setup
- bool MPTK_Load (byte[] datamidi, bool strict=false)
  Load Midi from an array of bytes
- bool MPTK_Load (string midiname, bool strict=false)
  Load Midi from a Midi file from Unity resources. The Midi file must be present in Unity MidiDB ressource folder.
- List< MPTKEvent > MPTK_ReadMidiEvents (long fromTicks=0, long toTicks=long.MaxValue)
  Read the list of midi events available in the Midi from a ticks position to an end position.
- double MPTK_ConvertTickToTime (long tick)
  Convert the tick duration to a real time duration in millisecond regarding the current tempo.
- long MPTK_ConvertTimeToTick (double time)
  Convert a real time duration in millisecond to a number of tick regarding the current tempo.

Public Attributes

- double MPTK_InitialTempo
  Initial tempo found in the Midi
- TimeSpan MPTK_Duration
  Duration of the midi. This duration is not constant depending of midi event change tempo inside the midi file.
- TimeSpan MPTK_RealDuration
  Real Duration of the midi calculated with the midi change tempo events find inside the midi file.
- long MPTK_TickLast
  Last tick position in Midi: Time of the last midi event in sequence expressed in number of "ticks". MPTK_TickLast / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.
- long MPTK_TickCurrent
  Current tick position in Midi: Time of the current midi event expressed in number of "ticks". MPTK_TickCurrent / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.
- int MPTK_NumberBeatsMeasure
  From TimeSignature event: The numerator counts the number of beats in a measure. For example a numerator of 4 means that each bar contains four beats. This is important to know because usually the first beat of each bar has extra emphasis. http://www.deluge.co/?q=midi-tempo-bpm
- int MPTK_NumberQuarterBeat
  From TimeSignature event: number of quarter notes in a beat. Equal 2 Power TimeSigDenominator. http://www.deluge.co/?q=midi-tempo-bpm
- int MPTK_TimeSigNumerator
  From TimeSignature event: The numerator counts the number of beats in a measure. For example a numerator of 4 means that each bar contains four beats. This is important to know because usually

the first beat of each bar has extra emphasis. In MIDI the denominator value is stored in a special format. i.e. the real denominator = 2^[dd] http://www.deluge.co/?q=midi-tempo-bpm

- int MPTK_TimeSigDenominator
  From TimeSignature event: The denominator specifies the number of quarter notes in a beat. 2 represents a quarter-note, 3 represents an eighth-note, etc. . http://www.deluge.co/?q=midi-tempo-bpm
- int MPTK_TicksInMetronomeClick
  From TimeSignature event: The standard MIDI clock ticks every 24 times every quarter note (crotchet) so a [cc] value of 24 would mean that the metronome clicks once every quarter note. A [cc] value of 6 would mean that the metronome clicks once every 1/8th of a note (quaver). http://www.deluge.co/?q=midi-tempo-bpm
- int MPTK_No32ndNotesInQuarterNote
  From TimeSignature event: This value specifies the number of 1/32nds of a note happen every MIDI quarter note. It is usually 8 which means that a quarter note happens every quarter note. http://www.deluge.co/?q=midi-tempo-bpm
- int MPTK_MicrosecondsPerQuarterNote
  From the SetTempo event: The tempo is given in micro seconds per quarter beat. To convert this to BPM we needs to use the following equation:BPM = 60,000,000/[tt tt tt] Warning: this value can change during the playing when a change tempo event is find. http://www.deluge.co/?q=midi-tempo-bpm
- int MPTK_DeltaTicksPerQuarterNote
  From Midi Header: Delta Ticks Per Quarter Note. Represent the duration time in "ticks" which make up a quarter-note. For instance, if 96, then a duration of an eighth-note in the file would be 48.
- int MPTK_TrackCount
  Count of track read in the Midi file

Detailed Description

Base class for loading a Midi file. No seqencer, no synthetizer. Usefull to load all the Midi events from a Midi.

```
 MidiLoad MidiLoaded = new MidiLoad();
 MidiLoaded.MPTK Load(midiindex);
 List<MPTKEvent> events = MidiLoaded.MPTK_ReadMidiEvents();
!
```

///

Member Function Documentation

bool MidiPlayerTK.MidiLoad.MPTK_Load (int *index*, bool *strict* = `false`)

Load Midi from midi MPTK referential (Unity resource). The index of the Midi file can be found in the windo "Midi File Setup". Display with menu MPTK / Midi File Setup

Parameters:

index	
strict	If true will error on non-paired note events, default:false

Returns:

true if loaded

bool MidiPlayerTK.MidiLoad.MPTK_Load (byte [] *datamidi*, bool *strict* = `false`)

Load Midi from an array of bytes

Parameters:

datamidi	byte arry midi
strict	If true will error on non-paired note events, default:false

Returns:

true if loaded

bool MidiPlayerTK.MidiLoad.MPTK_Load (string *midiname*, bool *strict* = `false`)

Load Midi from a Midi file from Unity resources. The Midi file must be present in Unity MidiDB ressource folder.

Parameters:

midiname	Midi file name without path and extension
strict	if true, check strict compliance with the Midi norm

Returns:

true if loaded

List<MPTKEvent> MidiPlayerTK.MidiLoad.MPTK_ReadMidiEvents (long *fromTicks* = `0`, long *toTicks* = `long.MaxValue`)

Read the list of midi events available in the Midi from a ticks position to an end position.

Parameters:

fromTicks	ticks start
toTicks	ticks end

Returns:

double MidiPlayerTK.MidiLoad.MPTK_ConvertTickToTime (long *tick*)

Convert the tick duration to a real time duration in millisecond regarding the current tempo.

Parameters:

tick	duration in ticks

Returns:

duration in milliseconds

long MidiPlayerTK.MidiLoad.MPTK_ConvertTimeToTick (double *time*)

Convert a real time duration in millisecond to a number of tick regarding the current tempo.

Parameters:

time	duration in milliseconds

Returns:

duration in ticks

Member Data Documentation

double MidiPlayerTK.MidiLoad.MPTK_InitialTempo

Initial tempo found in the Midi

TimeSpan MidiPlayerTK.MidiLoad.MPTK_Duration

Duration of the midi. This duration is not constant depending of midi event change tempo inside the midi file.

TimeSpan MidiPlayerTK.MidiLoad.MPTK_RealDuration

Real Duration of the midi calculated with the midi change tempo events find inside the midi file.

long MidiPlayerTK.MidiLoad.MPTK_TickLast

Last tick position in Midi: Time of the last midi event in sequence expressed in number of "ticks". MPTK_TickLast / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.

long MidiPlayerTK.MidiLoad.MPTK_TickCurrent

Current tick position in Midi: Time of the current midi event expressed in number of "ticks". MPTK_TickCurrent / MPTK_DeltaTicksPerQuarterNote equal the duration time of a quarter-note regardless the defined tempo.

int MidiPlayerTK.MidiLoad.MPTK_NumberBeatsMeasure

From TimeSignature event: The numerator counts the number of beats in a measure. For example a numerator of 4 means that each bar contains four beats. This is important to know because usually the first beat of each bar has extra emphasis. http://www.deluge.co/?q=midi-tempo-bpm

int MidiPlayerTK.MidiLoad.MPTK_NumberQuarterBeat

From TimeSignature event: number of quarter notes in a beat. Equal 2 Power TimeSigDenominator. http://www.deluge.co/?q=midi-tempo-bpm

int MidiPlayerTK.MidiLoad.MPTK_TimeSigNumerator

From TimeSignature event: The numerator counts the number of beats in a measure. For example a numerator of 4 means that each bar contains four beats. This is important to know because usually the first beat of each bar has extra emphasis. In MIDI the denominator value is stored in a special format. i.e. the real denominator = 2^[dd] http://www.deluge.co/?q=midi-tempo-bpm

int MidiPlayerTK.MidiLoad.MPTK_TimeSigDenominator

From TimeSignature event: The denominator specifies the number of quarter notes in a beat. 2 represents a quarter-note, 3 represents an eighth-note, etc. . http://www.deluge.co/?q=midi-tempo-bpm

int MidiPlayerTK.MidiLoad.MPTK_TicksInMetronomeClick

From TimeSignature event: The standard MIDI clock ticks every 24 times every quarter note (crotchet) so a [cc] value of 24 would mean that the metronome clicks once every quarter note. A [cc] value of 6 would mean that the metronome clicks once every 1/8th of a note (quaver). http://www.deluge.co/?q=midi-tempo-bpm

int MidiPlayerTK.MidiLoad.MPTK_No32ndNotesInQuarterNote

From TimeSignature event: This value specifies the number of 1/32nds of a note happen every MIDI quarter note. It is usually 8 which means that a quarter note happens every quarter note. http://www.deluge.co/?q=midi-tempo-bpm

int MidiPlayerTK.MidiLoad.MPTK_MicrosecondsPerQuarterNote

From the SetTempo event: The tempo is given in micro seconds per quarter beat. To convert this to BPM we needs to use the following equation:BPM = 60,000,000/[tt tt tt] Warning: this value can change during the playing when a change tempo event is find. http://www.deluge.co/?q=midi-tempo-bpm

int MidiPlayerTK.MidiLoad.MPTK_DeltaTicksPerQuarterNote

From Midi Header: Delta Ticks Per Quarter Note. Represent the duration time in "ticks" which make up a quarter-note. For instance, if 96, then a duration of an eighth-note in the file would be 48.

int MidiPlayerTK.MidiLoad.MPTK_TrackCount

Count of track read in the Midi file

MidiPlayerTK.MidiPlayerGlobal

Singleton class to manage all global features of MPTK.
Inherits MonoBehaviour.

Static Public Member Functions

- static bool MPTK_IsReady (float delay=0.5f)
  Check if SoudFont is loaded. Add a default wait time because Unity AudioSource need a delay to be really ready to play. Hummm, like a diesel motor ?
- static void MPTK_SelectSoundFont (string name)
  Changing the current Soundfont on fly. If some Midis are playing they are restarted.
- static void MPTK_SelectBankInstrument (int nbank)
  Change default current bank on fly
- static void MPTK_SelectBankDrum (int nbank)
  Change current bank on fly
- static int MPTK_FindMidi (string name)
  Find index of a Midi by name. Use the exact name defined in Unity resources folder MidiDB without any path or extension. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK.
- static float MPTK_DistanceToListener (Transform trf)
  Calculate distance with the AudioListener.

Static Public Attributes

- static string MPTK_PathToResources = "MidiPlayer/Resources/"
  This path could change depending your project. Change the path before any actions in MPTK.
- static int MPTK_CountWaveLoaded
  Count of wave loaded
- static bool MPTK_SoundFontLoaded = false
  True if soundfont is loaded
- static List< MPTKListItem > MPTK_ListMidi
  List of midi(s) available
- static List< MPTKListItem > MPTK_ListPreset
  Get the list of presets available for instruments for the selected bank
- static List< MPTKListItem > MPTK_ListBank
  Get the list of banks available
- static List< MPTKListItem > MPTK_ListPresetDrum
  Get the list of presets available for instrument
- static List< MPTKListItem > MPTK_ListDrum
  Get the list of presets available

Properties

- static TimeSpan MPTK_TimeToLoadSoundFont `[get]`
  Load time for the current SoundFont
- static TimeSpan MPTK_TimeToLoadWave `[get]`
  Load time for the wave
- static int MPTK_CountPresetLoaded `[get]`
  Count of preset loaded
- static UnityEvent OnEventPresetLoaded `[get, set]`
  Event triggered at end of loading a soundfont. Warning: when defined by script, this event is not triggered at first load of MPTK because MidiPlayerGlobal is loaded before any other gamecomponent. Set this event in the Inspector of MidiPlayerGlobal to get at first load this information.
- static List< string > MPTK_ListSoundFont `[get]`
  List of Soundfont(s) available

Detailed Description

Singleton class to manage all global features of MPTK.

Member Function Documentation

static bool MidiPlayerTK.MidiPlayerGlobal.MPTK_IsReady (float *delay* = 0.5f) `[static]`

Check if SoudFont is loaded. Add a default wait time because Unity AudioSource need a delay to be really ready to play. Hummm, like a diesel motor ?

Parameters:

delay	

Returns:

static void MidiPlayerTK.MidiPlayerGlobal.MPTK_SelectSoundFont (string *name*) `[static]`

Changing the current Soundfont on fly. If some Midis are playing they are restarted.

Parameters:

name	SoundFont name

static void MidiPlayerTK.MidiPlayerGlobal.MPTK_SelectBankInstrument (int *nbank*) `[static]`

Change default current bank on fly

Parameters:

nbank	Number of the SoundFont Bank to load for instrument.

static void MidiPlayerTK.MidiPlayerGlobal.MPTK_SelectBankDrum (int *nbank*) `[static]`

Change current bank on fly

Parameters:

nbank	Number of the SoundFont Bank to load for drum.

static int MidiPlayerTK.MidiPlayerGlobal.MPTK_FindMidi (string *name*) `[static]`

Find index of a Midi by name. Use the exact name defined in Unity resources folder MidiDB without any path or extension. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK.

Parameters:

name	name of the midi without path nor extension

Returns:

-1 if not found else return the index of the midi.

static float MidiPlayerTK.MidiPlayerGlobal.MPTK_DistanceToListener (Transform *trf*) `[static]`

Calculate distance with the AudioListener.

Parameters:

trf	Transform of the object to calculate the distance.

Returns:

Member Data Documentation

string MidiPlayerTK.MidiPlayerGlobal.MPTK_PathToResources = "MidiPlayer/Resources/" `[static]`

This path could change depending your project. Change the path before any actions in MPTK.

int MidiPlayerTK.MidiPlayerGlobal.MPTK_CountWaveLoaded `[static]`

Count of wave loaded

bool MidiPlayerTK.MidiPlayerGlobal.MPTK_SoundFontLoaded = false `[static]`

True if soundfont is loaded

List<MPTKListItem> MidiPlayerTK.MidiPlayerGlobal.MPTK_ListMidi [static]

List of midi(s) available

List<MPTKListItem> MidiPlayerTK.MidiPlayerGlobal.MPTK_ListPreset [static]

Get the list of presets available for instruments for the selected bank

List<MPTKListItem> MidiPlayerTK.MidiPlayerGlobal.MPTK_ListBank [static]

Get the list of banks available

List<MPTKListItem> MidiPlayerTK.MidiPlayerGlobal.MPTK_ListPresetDrum [static]

Get the list of presets available for instrument

List<MPTKListItem> MidiPlayerTK.MidiPlayerGlobal.MPTK_ListDrum [static]

Get the list of presets available

Property Documentation

TimeSpan MidiPlayerTK.MidiPlayerGlobal.MPTK_TimeToLoadSoundFont [static], [get]

Load time for the current SoundFont

TimeSpan MidiPlayerTK.MidiPlayerGlobal.MPTK_TimeToLoadWave [static], [get]

Load time for the wave

int MidiPlayerTK.MidiPlayerGlobal.MPTK_CountPresetLoaded [static], [get]

Count of preset loaded

UnityEvent MidiPlayerTK.MidiPlayerGlobal.OnEventPresetLoaded `[static], [get], [set]`

Event triggered at end of loading a soundfont. Warning: when defined by script, this event is not triggered at first load of MPTK because MidiPlayerGlobal is loaded before any other gamecomponent. Set this event in the Inspector of MidiPlayerGlobal to get at first load this information.

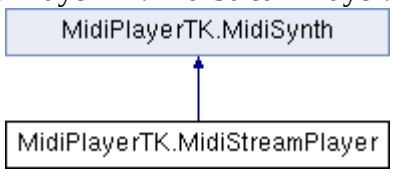
List<string> MidiPlayerTK.MidiPlayerGlobal.MPTK_ListSoundFont `[static], [get]`

List of Soundfont(s) available

MidiPlayerTK.MidiStreamPlayer

Play generated notes. Any Midi file is necessary rather create music from your own algorithm with MPTK_PlayEvent(). Duration can be set in the MPTKEvent, but a note can also be stopped with MPTK_StopEvent().

Inheritance diagram for MidiPlayerTK.MidiStreamPlayer:



Public Member Functions

- void MPTK_PlayEvent (MPTKEvent evnt)
  Play one midi event with a thread so the call return immediately.

```
midiStreamPlayer.MPTK_PlayEvent
(
    new MPTKEvent()
    {
        Channel = 9,
        Duration = 999999,
        Value = 48,
        Velocity = 100
    }
);
```

- void MPTK_PlayEvent (List< MPTKEvent > events)
  Play a list of midi events with a thread so the call return immediately.

```
        void Update()
        {
            // Checj that SoundFont is loaded and add a little wait (0.5 s by
default) because Unity AudioSource need some time to be started
            if (!MidiPlayerGlobal.MPTK IsReady())
                return;

            if (DrumKit)
                // Set canal to dedicated drum canal (9 if canal start from 0,
canal 10 is displayed in log)
                StreamChannel = 9;
            else
                StreamChannel = 0;
```

```
            if (midiStreamPlayer != null && (IsplayingLoopPresets ||
IsplayingLoopNotes))
            {
                float time = Time.realtimeSinceStartup - LastTimeChange;
                if (time > DelayTimeChange)
                {
                    // It's time to generate some notes ;-)
                    LastTimeChange = Time.realtimeSinceStartup;

                    int noteToPlay;
                    if (ChordPlay) noteToPlay = 3;
                    else if (ArpeggioPlay) noteToPlay = 5;
                    else noteToPlay = 1;

                    while (--noteToPlay >= 0)
                    {
                        if (IsplayingLoopPresets)
                        {
                            if (++CurrentPreset > EndPreset) CurrentPreset =
StartPreset;
                            if (CurrentPreset < StartPreset) CurrentPreset =
StartPreset;
                            midiStreamPlayer.MPTK_PlayEvent(new MPTKEvent() {
Command = MPTKCommand.PatchChange, Value = CurrentPreset, Channel = StreamChannel,
});
                        }

                        if (IsplayingLoopNotes)
                        {
                            if (++CurrentNote > EndNote) CurrentNote = StartNote;
                            if (CurrentNote < StartNote) CurrentNote = StartNote;
                        }

                        // For chrod or arpeggiator
                        if (noteToPlay == 1)
                            CurrentNote += 3;
                        else if (noteToPlay == 2)
                            CurrentNote += 7;

                        if (RandomPlay)
                        {
                            CurrentNote += UnityEngine.Random.Range(-8, 8);
                        }

                        //Debug.LogFormat("Play {0}", CurrentNote);

                        long duration = Convert.ToInt64((DelayTimeChange +
DeltaDelay) * 1000);
                        if (duration < 0) duration = 1;

                        // Send the note to the player. Notes are plays in a
thread, so call returns immediately
                        // Note is stopped automatically avec the Duration
defined.
                        midiStreamPlayer.MPTK_PlayEvent(
                            new MPTKEvent()
                            {
                                Command = MPTKCommand.NoteOn,
                                Value = CurrentNote,
                                Channel = StreamChannel,
                                Duration = duration,
                                Velocity = Velocity,
                            });
                    }
                }
            }
        }
```

- void MPTK_StopEvent (MPTKEvent pnote)
  Stop playing the note. All waves associated to the note are stop by sending a noteoff.
- void MPTK_InitSynth (int channelCount=16)

Init the synthetizer. Prefabs automatically initialize the synthetizer (see events). It's not usefull to call this method.

- void MPTK_ClearAllSound (bool destroyAudioSource=false)
  Clear all sound

Public Attributes

- bool MPTK_CorePlayer
  If true then Midi events are read and play from a dedicated thread. If false, MidiSynth will use AudioSource gameobject to play sound. This properties must be defined before running the application from the inspector. The default is true.
- bool MPTK_DirectSendToPlayer
  If true (default) then Midi events are sent automatically to the midi player. Set to false if you want to process events without playing sound. OnEventNotesMidi Unity Event can be used to process each notes.
- bool MPTK_EnableChangeTempo
  Should accept change tempo from Midi Events ?
- bool MPTK_PauseOnDistance
  Should the Midi playing must be paused if distance between AudioListener and MidiFilePlayer is greater than MaxDistance
- bool MPTK_EnablePanChange
  Should change pan from Midi Events or from SoundFont ?
- bool MPTK_EnablePresetDrum
  Should accept change Preset for Drum canal 10 ? Disabled by default. Could sometimes create bad sound with midi files not really compliant with the Midi norm.
- bool MPTK_LogWave
  Log for each wave to be played
- uint MPTK_ReleaseTimeMin = 500000
  [Only when CorePlayer=False] Define a minimum release time at noteoff in 100 iem nanoseconds. Default 50 ms is a good tradeoff. Below some unpleasant sound could be heard. Useless when MPTK_CorePlayer is true.
- int MPTK_StatVoiceCountActive
  Count of the active voices (playing) - Readonly
- int MPTK_StatVoiceCountFree
  Count of the free voices for reusing on need. Older than AutoCleanVoiceTime are removed when count is over than AutoCleanVoiceLimit - Readonly
- float MPTK_StatVoiceRatioReused
  Percentage of voice reused during the synth life. 0: any reuse, 100:all voice reused (unattainable, of course!)
- int MPTK_StatVoicePlayed
  Count of voice played since the start of the synth
- int MPTK_AutoCleanVoiceLimit
  Free voices older than MPTK_AutoCleanVoiceLimit are removed when count is over than MPTK_AutoCleanVoiceTime
- bool MPTK_WeakDevice
  Should play on a weak device (cheaper smartphone) ? Apply only with AudioSource mode (MPTK_CorePlayer=False) Playing Midi files with WeakDevice activated could cause some bad interpretation of Midi Event, consequently bad sound.
- EventSynthClass OnEventSynthAwake
  Unity event fired at awake of the synthesizer. Name of the gameobject component is passed as a parameter.
- EventSynthClass OnEventSynthStarted

Unity event fired at start of the synthesizer. Name of the gameobject component is passed as a parameter.

Properties

- int MPTK_IndexSynthRate `[get, set]`
  Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.
- int MPTK_IndexSynthBuffSize `[get, set]`
  Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.
- float MPTK_MaxDistance `[get, set]`
  MaxDistance to use for PauseOnDistance
- float MPTK_Volume `[get, set]`
  Volume of midi playing. Must be >=0 and <= 1
- int MPTK_Transpose `[get, set]`
  Transpose note from -24 to 24

Detailed Description

Play generated notes. Any Midi file is necessary rather create music from your own algorithm with MPTK_PlayEvent(). Duration can be set in the MPTKEvent, but a note can also be stopped with MPTK_StopEvent().

Member Function Documentation

void MidiPlayerTK.MidiStreamPlayer.MPTK_PlayEvent (MPTKEvent *evnt*)

Play one midi event with a thread so the call return immediately.

```
midiStreamPlayer.MPTK_PlayEvent
(
    new MPTKEvent()
    {
        Channel = 9,
        Duration = 999999,
        Value = 48,
        Velocity = 100
    }
);
```

void MidiPlayerTK.MidiStreamPlayer.MPTK_PlayEvent (List< MPTKEvent > *events*)

Play a list of midi events with a thread so the call return immediately.

```
        void Update()
        {
            // Checj that SoundFont is loaded and add a little wait (0.5 s by
default) because Unity AudioSource need some time to be started
            if (!MidiPlayerGlobal.MPTK_IsReady())
                return;

            if (DrumKit)
```

```
                // Set canal to dedicated drum canal (9 if canal start from 0,
canal 10 is displayed in log)
                StreamChannel = 9;
            else
                StreamChannel = 0;

            if (midiStreamPlayer != null && (IsplayingLoopPresets ||
IsplayingLoopNotes))
            {
                float time = Time.realtimeSinceStartup - LastTimeChange;
                if (time > DelayTimeChange)
                {
                    // It's time to generate some notes ;-)
                    LastTimeChange = Time.realtimeSinceStartup;

                    int noteToPlay;
                    if (ChordPlay) noteToPlay = 3;
                    else if (ArpeggioPlay) noteToPlay = 5;
                    else noteToPlay = 1;

                    while (--noteToPlay >= 0)
                    {
                        if (IsplayingLoopPresets)
                        {
                            if (++CurrentPreset > EndPreset) CurrentPreset =
StartPreset;
                            if (CurrentPreset < StartPreset) CurrentPreset =
StartPreset;
                            midiStreamPlayer.MPTK_PlayEvent(new MPTKEvent() {
Command = MPTKCommand.PatchChange, Value = CurrentPreset, Channel = StreamChannel,
});
                        }

                        if (IsplayingLoopNotes)
                        {
                            if (++CurrentNote > EndNote) CurrentNote = StartNote;
                            if (CurrentNote < StartNote) CurrentNote = StartNote;
                        }

                        // For chrod or arpeggiator
                        if (noteToPlay == 1)
                            CurrentNote += 3;
                        else if (noteToPlay == 2)
                            CurrentNote += 7;

                        if (RandomPlay)
                        {
                            CurrentNote += UnityEngine.Random.Range(-8, 8);
                        }

                        //Debug.LogFormat("Play {0}", CurrentNote);

                        long duration = Convert.ToInt64((DelayTimeChange +
DeltaDelay) * 1000);
                        if (duration < 0) duration = 1;

                        // Send the note to the player. Notes are plays in a
thread, so call returns immediately
                        // Note is stopped automatically avec the Duration
defined.
                        midiStreamPlayer.MPTK PlayEvent(
                            new MPTKEvent()
                            {
                                Command = MPTKCommand.NoteOn,
                                Value = CurrentNote,
                                Channel = StreamChannel,
                                Duration = duration,
                                Velocity = Velocity,
                            });
                    }
                }
            }
        }
```

void MidiPlayerTK.MidiStreamPlayer.MPTK_StopEvent (MPTKEvent *pnote*)

Stop playing the note. All waves associated to the note are stop by sending a noteoff.

Parameters:

pnote	

void MidiPlayerTK.MidiSynth.MPTK_InitSynth (int *channelCount* = 16) [inherited]

Init the synthetizer. Prefabs automatically initialize the synthetizer (see events). It's not usefull to call this method.

Parameters:

channelCount	Number of channel to create, default 16. Any other values are experimental!

void MidiPlayerTK.MidiSynth.MPTK_ClearAllSound (bool *destroyAudioSource* = false) [inherited]

Clear all sound

Parameters:

destroyAudioSource	Destroy also audioSource (default:false)

```
  if (GUILayout.Button("Clear"))
     midiStreamPlayer.MPTK_ClearAllSound(true);
!
```

Member Data Documentation

bool MidiPlayerTK.MidiSynth.MPTK_CorePlayer [inherited]

If true then Midi events are read and play from a dedicated thread. If false, MidiSynth will use AudioSource gameobject to play sound. This properties must be defined before running the application from the inspector. The default is true.

bool MidiPlayerTK.MidiSynth.MPTK_DirectSendToPlayer [inherited]

If true (default) then Midi events are sent automatically to the midi player. Set to false if you want to process events without playing sound. OnEventNotesMidi Unity Event can be used to process each notes.

bool MidiPlayerTK.MidiSynth.MPTK_EnableChangeTempo [inherited]

Should accept change tempo from Midi Events ?

bool MidiPlayerTK.MidiSynth.MPTK_PauseOnDistance [inherited]

Should the Midi playing must be paused if distance between AudioListener and MidiFilePlayer is greater than MaxDistance

bool MidiPlayerTK.MidiSynth.MPTK_EnablePanChange [inherited]

Should change pan from Midi Events or from SoundFont ?

bool MidiPlayerTK.MidiSynth.MPTK_EnablePresetDrum [inherited]

Should accept change Preset for Drum canal 10 ? Disabled by default. Could sometimes create bad sound with midi files not really compliant with the Midi norm.

bool MidiPlayerTK.MidiSynth.MPTK_LogWave [inherited]

Log for each wave to be played

uint MidiPlayerTK.MidiSynth.MPTK_ReleaseTimeMin = 500000 [inherited]

[Only when CorePlayer=False] Define a minimum release time at noteoff in 100 iem nanoseconds. Default 50 ms is a good tradeoff. Below some unpleasant sound could be heard. Useless when MPTK_CorePlayer is true.

int MidiPlayerTK.MidiSynth.MPTK_StatVoiceCountActive [inherited]

Count of the active voices (playing) - Readonly

int MidiPlayerTK.MidiSynth.MPTK_StatVoiceCountFree [inherited]

Count of the free voices for reusing on need. Older than AutoCleanVoiceTime are removed when count is over than AutoCleanVoiceLimit - Readonly

float MidiPlayerTK.MidiSynth.MPTK_StatVoiceRatioReused [inherited]

Percentage of voice reused during the synth life. 0: any reuse, 100:all voice reused (unattainable, of course!)

int MidiPlayerTK.MidiSynth.MPTK_StatVoicePlayed [inherited]

Count of voice played since the start of the synth

int MidiPlayerTK.MidiSynth.MPTK_AutoCleanVoiceLimit [inherited]

Free voices older than MPTK_AutoCleanVoiceLimit are removed when count is over than MPTK_AutoCleanVoiceTime

bool MidiPlayerTK.MidiSynth.MPTK_WeakDevice [inherited]

Should play on a weak device (cheaper smartphone) ? Apply only with AudioSource mode (MPTK_CorePlayer=False) Playing Midi files with WeakDevice activated could cause some bad interpretation of Midi Event, consequently bad sound.

EventSynthClass MidiPlayerTK.MidiSynth.OnEventSynthAwake [inherited]

Unity event fired at awake of the synthesizer. Name of the gameobject component is passed as a parameter.

```
 ...
 if (!midiStreamPlayer.OnEventSynthAwake.HasEvent())
    midiStreamPlayer.OnEventSynthAwake.AddListener(StartLoadingSynth);
 ...
 public void StartLoadingSynth(string name)
 {
     Debug.LogFormat("Synth {0} loading", name);
 }
!
```

EventSynthClass MidiPlayerTK.MidiSynth.OnEventSynthStarted [inherited]

Unity event fired at start of the synthesizer. Name of the gameobject component is passed as a parameter.

```
 ...
 if (!midiStreamPlayer.OnEventStartSynth.HasEvent())
    midiStreamPlayer.OnEventStartSynth.AddListener(EndLoadingSynth);
 ...
 public void EndLoadingSynth(string name)
 {
    Debug.LogFormat("Synth {0} loaded", name);
    midiStreamPlayer.MPTK_PlayEvent(
```

```
       new MPTKEvent() { Command = MPTKCommand.PatchChange, Value =
CurrentPatchInstrument, Channel = StreamChannel});
 }
!
```

Property Documentation

int MidiPlayerTK.MidiSynth.MPTK_IndexSynthRate `[get], [set], [inherited]`

Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.

int MidiPlayerTK.MidiSynth.MPTK_IndexSynthBuffSize `[get], [set], [inherited]`

Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.

float MidiPlayerTK.MidiSynth.MPTK_MaxDistance `[get], [set], [inherited]`

MaxDistance to use for PauseOnDistance

float MidiPlayerTK.MidiSynth.MPTK_Volume `[get], [set], [inherited]`

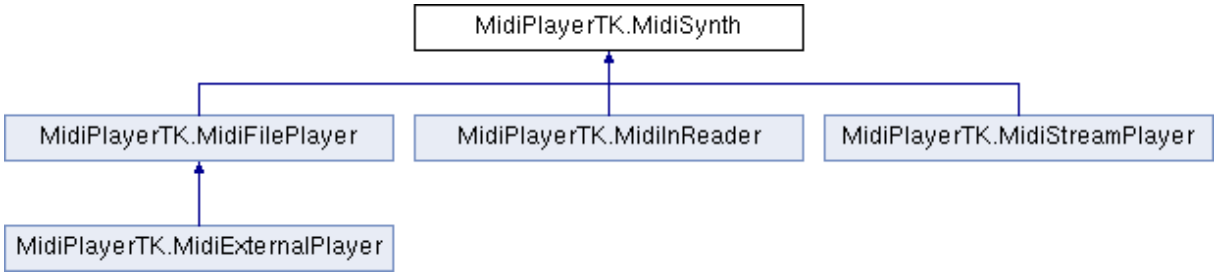
Volume of midi playing. Must be >=0 and <= 1

int MidiPlayerTK.MidiSynth.MPTK_Transpose `[get], [set], [inherited]`

Transpose note from -24 to 24

MidiPlayerTK.MidiSynth

Inheritance diagram for MidiPlayerTK.MidiSynth:



Public Member Functions

- void Awake ()
  From fluid_sys.c - fluid_utime() returns the time in micro seconds. this time should only be used to measure duration(relative times).
- void Start ()
- void MPTK_InitSynth (int channelCount=16)
  Init the synthetizer. Prefabs automatically initialize the synthetizer (see events). It's not usefull to call this method.
- void MPTK_ClearAllSound (bool destroyAudioSource=false)
  Clear all sound

Public Attributes

- bool MPTK_CorePlayer
  If true then Midi events are read and play from a dedicated thread. If false, MidiSynth will use AudioSource gameobject to play sound. This properties must be defined before running the application from the inspector. The default is true.
- bool MPTK_DirectSendToPlayer
  If true (default) then Midi events are sent automatically to the midi player. Set to false if you want to process events without playing sound. OnEventNotesMidi Unity Event can be used to process each notes.
- bool MPTK_EnableChangeTempo
  Should accept change tempo from Midi Events ?
- bool MPTK_PauseOnDistance
  Should the Midi playing must be paused if distance between AudioListener and MidiFilePlayer is greater than MaxDistance
- bool MPTK_EnablePanChange
  Should change pan from Midi Events or from SoundFont ?
- bool MPTK_EnablePresetDrum
  Should accept change Preset for Drum canal 10 ? Disabled by default. Could sometimes create bad sound with midi files not really compliant with the Midi norm.
- bool MPTK_LogWave
  Log for each wave to be played
- uint MPTK_ReleaseTimeMin = 500000
  [Only when CorePlayer=False] Define a minimum release time at noteoff in 100 iem nanoseconds. Default 50 ms is a good tradeoff. Below some unpleasant sound could be heard. Useless when MPTK_CorePlayer is true.
- int MPTK_StatVoiceCountActive
  Count of the active voices (playing) - Readonly
- int MPTK_StatVoiceCountFree
  Count of the free voices for reusing on need. Older than AutoCleanVoiceTime are removed when count is over than AutoCleanVoiceLimit - Readonly
- float MPTK_StatVoiceRatioReused

Percentage of voice reused during the synth life. 0: any reuse, 100:all voice reused (unattainable, of course!)

- int MPTK_StatVoicePlayed
  Count of voice played since the start of the synth
- int MPTK_AutoCleanVoiceLimit
  Free voices older than MPTK_AutoCleanVoiceLimit are removed when count is over than MPTK_AutoCleanVoiceTime
- bool MPTK_WeakDevice
  Should play on a weak device (cheaper smartphone) ? Apply only with AudioSource mode (MPTK_CorePlayer=False) Playing Midi files with WeakDevice activated could cause some bad interpretation of Midi Event, consequently bad sound.
- EventSynthClass OnEventSynthAwake
  Unity event fired at awake of the synthesizer. Name of the gameobject component is passed as a parameter.
- EventSynthClass OnEventSynthStarted
  Unity event fired at start of the synthesizer. Name of the gameobject component is passed as a parameter.

Properties

- int MPTK_IndexSynthRate `[get, set]`
  Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.
- int MPTK_IndexSynthBuffSize `[get, set]`
  Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.
- float MPTK_MaxDistance `[get, set]`
  MaxDistance to use for PauseOnDistance
- float MPTK_Volume `[get, set]`
  Volume of midi playing. Must be >=0 and <= 1
- int MPTK_Transpose `[get, set]`
  Transpose note from -24 to 24

Detailed Description

Base class for Midi Synthesizer. Migrated from fluidsynth. It's not recommended to instanciate this class. Instead use MidiFilePlayer or MidiStreamPlayer.

Member Function Documentation

void MidiPlayerTK.MidiSynth.Awake ()

From fluid_sys.c - fluid_utime() returns the time in micro seconds. this time should only be used to measure duration(relative times).

Returns:

returns the current time in milliseconds. This time should only be used in relative time measurements.

Returns:

void MidiPlayerTK.MidiSynth.Start ()

- Allocate the sample buffers */
- Left and right audio buffers */
- Effects audio buffers */
- allocate the reverb module */
- allocate the chorus module */

void MidiPlayerTK.MidiSynth.MPTK_InitSynth (int *channelCount* = 16)

Init the synthetizer. Prefabs automatically initialize the synthetizer (see events). It's not usefull to call this method.

Parameters:

channelCount	Number of channel to create, default 16. Any other values are experimental!

void MidiPlayerTK.MidiSynth.MPTK_ClearAllSound (bool *destroyAudioSource* = false)

Clear all sound

Parameters:

destroyAudioSource	Destroy also audioSource (default:false)

```
  if (GUILayout.Button("Clear"))
     midiStreamPlayer.MPTK_ClearAllSound(true);
!
```

Member Data Documentation

bool MidiPlayerTK.MidiSynth.MPTK_CorePlayer

If true then Midi events are read and play from a dedicated thread. If false, MidiSynth will use AudioSource gameobject to play sound. This properties must be defined before running the application from the inspector. The default is true.

bool MidiPlayerTK.MidiSynth.MPTK_DirectSendToPlayer

If true (default) then Midi events are sent automatically to the midi player. Set to false if you want to process events without playing sound. OnEventNotesMidi Unity Event can be used to process each notes.

bool MidiPlayerTK.MidiSynth.MPTK_EnableChangeTempo

Should accept change tempo from Midi Events ?

bool MidiPlayerTK.MidiSynth.MPTK_PauseOnDistance

Should the Midi playing must be paused if distance between AudioListener and MidiFilePlayer is greater than MaxDistance

bool MidiPlayerTK.MidiSynth.MPTK_EnablePanChange

Should change pan from Midi Events or from SoundFont ?

bool MidiPlayerTK.MidiSynth.MPTK_EnablePresetDrum

Should accept change Preset for Drum canal 10 ? Disabled by default. Could sometimes create bad sound with midi files not really compliant with the Midi norm.

bool MidiPlayerTK.MidiSynth.MPTK_LogWave

Log for each wave to be played

uint MidiPlayerTK.MidiSynth.MPTK_ReleaseTimeMin = 500000

[Only when CorePlayer=False] Define a minimum release time at noteoff in 100 iem nanoseconds. Default 50 ms is a good tradeoff. Below some unpleasant sound could be heard. Useless when MPTK_CorePlayer is true.

int MidiPlayerTK.MidiSynth.MPTK_StatVoiceCountActive

Count of the active voices (playing) - Readonly

int MidiPlayerTK.MidiSynth.MPTK_StatVoiceCountFree

Count of the free voices for reusing on need. Older than AutoCleanVoiceTime are removed when count is over than AutoCleanVoiceLimit - Readonly

float MidiPlayerTK.MidiSynth.MPTK_StatVoiceRatioReused

Percentage of voice reused during the synth life. 0: any reuse, 100:all voice reused (unattainable, of course!)

int MidiPlayerTK.MidiSynth.MPTK_StatVoicePlayed

Count of voice played since the start of the synth

int MidiPlayerTK.MidiSynth.MPTK_AutoCleanVoiceLimit

Free voices older than MPTK_AutoCleanVoiceLimit are removed when count is over than MPTK_AutoCleanVoiceTime

bool MidiPlayerTK.MidiSynth.MPTK_WeakDevice

Should play on a weak device (cheaper smartphone) ? Apply only with AudioSource mode (MPTK_CorePlayer=False) Playing Midi files with WeakDevice activated could cause some bad interpretation of Midi Event, consequently bad sound.

EventSynthClass MidiPlayerTK.MidiSynth.OnEventSynthAwake

Unity event fired at awake of the synthesizer. Name of the gameobject component is passed as a parameter.

```
...
if (!midiStreamPlayer.OnEventSynthAwake.HasEvent())
   midiStreamPlayer.OnEventSynthAwake.AddListener(StartLoadingSynth);
...
public void StartLoadingSynth(string name)
{
    Debug.LogFormat("Synth {0} loading", name);
}
!
```

EventSynthClass MidiPlayerTK.MidiSynth.OnEventSynthStarted

Unity event fired at start of the synthesizer. Name of the gameobject component is passed as a parameter.

```
...
if (!midiStreamPlayer.OnEventStartSynth.HasEvent())
   midiStreamPlayer.OnEventStartSynth.AddListener(EndLoadingSynth);
...
public void EndLoadingSynth(string name)
{
```

```
    Debug.LogFormat("Synth {0} loaded", name);
    midiStreamPlayer.MPTK PlayEvent(
       new MPTKEvent() { Command = MPTKCommand.PatchChange, Value =
CurrentPatchInstrument, Channel = StreamChannel});
 }
!
```

Property Documentation

int MidiPlayerTK.MidiSynth.MPTK_IndexSynthRate [get], [set]

Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.

int MidiPlayerTK.MidiSynth.MPTK_IndexSynthBuffSize [get], [set]

Set or Get sample rate output of the synth. -1:default, 0:24000, 1:36000, 2:48000, 3:60000, 4:72000, 5:84000, 6:96000. It's better to stop playing before changing on fly to avoid bad noise.

float MidiPlayerTK.MidiSynth.MPTK_MaxDistance [get], [set]

MaxDistance to use for PauseOnDistance

float MidiPlayerTK.MidiSynth.MPTK_Volume [get], [set]

Volume of midi playing. Must be >=0 and <= 1

int MidiPlayerTK.MidiSynth.MPTK_Transpose [get], [set]

Transpose note from -24 to 24

MidiPlayerTK.MidiListPlayer.MPTK_MidiPlayItem

Define a midi to be added in the list

Public Attributes

- string MidiName
  Midi Name. Use the exact name defined in Unity resources folder MidiDB without any path or extension. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK.

- bool UIAction
  Select or unselect this Midi in the Inspector to apply actions (reorder, delete, ...) NO MORE USED
- bool Selected
  Select or unselect this Midi to be played in the list ...)
- int Index
  Position of the Midi in the list. Use method MPTK_ReIndexMidi() recalculate the index.
- float StartFrom
  Time (ms) position where to start playing the midi file
- float EndFrom
  Time (ms) position where to end playing the midi file

Detailed Description

Define a midi to be added in the list

Member Data Documentation

string MidiPlayerTK.MidiListPlayer.MPTK_MidiPlayItem.MidiName

Midi Name. Use the exact name defined in Unity resources folder MidiDB without any path or extension. Tips: Add Midi files to your project with the Unity menu MPTK or add it directly in the ressource folder and open Midi File Setup to automatically integrate Midi in MPTK.

bool MidiPlayerTK.MidiListPlayer.MPTK_MidiPlayItem.UIAction

Select or unselect this Midi in the Inspector to apply actions (reorder, delete, ...) NO MORE USED

bool MidiPlayerTK.MidiListPlayer.MPTK_MidiPlayItem.Selected

Select or unselect this Midi to be played in the list ...)

int MidiPlayerTK.MidiListPlayer.MPTK_MidiPlayItem.Index

Position of the Midi in the list. Use method MPTK_ReIndexMidi() recalculate the index.

float MidiPlayerTK.MidiListPlayer.MPTK_MidiPlayItem.StartFrom

Time (ms) position where to start playing the midi file

float MidiPlayerTK.MidiListPlayer.MPTK_MidiPlayItem.EndFrom

Time (ms) position where to end playing the midi file

MidiPlayerTK.MPTKEvent

Midi Event class for MPTK. Usage to generate Midi Music with MidiStreamPlayer or to read midi events from a Midi file with MidiLoad or to recevice midi events from MidiFilePlayer OnEventNotesMidi.

Public Types

- enum EnumLength

Note length as https://en.wikipedia.org/wiki/Note_value Public Member Functions

- MPTKEvent (ulong data)
  Create a MPTK Midi event from a midi input message
- void Play (MidiStreamPlayer streamPlayer)
  Play a note which is stoppable. DEPRECATED in V2. Replaced by MPTK_PlayEvent in MidiStreamPlayer.
- void Stop ()
  Stop the note. DEPRECATED in V2. Replaced by MPTK_StopEvent in MidiStreamPlayer.
- override string ToString ()
  Build a string description of the Midi event.

Public Attributes

- long Track
  Index of track.
- long Tick
  Time in Midi Tick (part of a Beat) of the Event since the start of playing the midi file. This time is independant of the Tempo or Speed. Not used for MidiStreamPlayer.
- MPTKCommand Command
  Midi Command code. Defined the type of message (Note On, Control Change, Patch Change...)
- MPTKController Controller
  Controller code. When the Command is ControlChange, contains the code fo the controller to change (Modulation, Pan, Bank Select ...). Value will contains the value of the controller.
- MPTKMeta Meta
  MetaEvent Code. When the Command is MetaEvent, contains the code of the meta event (Lyric, TimeSignature, ...). . Info will contains the value of the meta.
- string Info
  Information hold by textual meta event when Command=MetaEvent
- int Value
  Contains a value between 0 and 127 in relation with the Command. For:
- int Channel
  Midi channel fom 0 to 15 (9 for drum)
- int Velocity

Velocity between 0 and 127

- long Duration
  Duration of the note in millisecond
- int Length
  Duration of the note in Midi Tick. MidiFilePlayer.MPTK_NoteLength can be used to convert this duration. Not used for MidiStreamPlayer. https://en.wikipedia.org/wiki/Note_value
- List< fluid_voice > Voices
  List of voices associated to this Event for playing a NoteOn event.

Detailed Description

Midi Event class for MPTK. Usage to generate Midi Music with MidiStreamPlayer or to read midi events from a Midi file with MidiLoad or to recevice midi events from MidiFilePlayer OnEventNotesMidi.

Member Enumeration Documentation

enum MidiPlayerTK.MPTKEvent.EnumLength [strong]

Note length as https://en.wikipedia.org/wiki/Note_value

Constructor & Destructor Documentation

MidiPlayerTK.MPTKEvent.MPTKEvent (ulong *data*)

Create a MPTK Midi event from a midi input message

Parameters:

data	

Member Function Documentation

void MidiPlayerTK.MPTKEvent.Play (MidiStreamPlayer *streamPlayer*)

Play a note which is stoppable. DEPRECATED in V2. Replaced by MPTK_PlayEvent in MidiStreamPlayer.

Parameters:

streamPlayer	A MidiStreamPlayer component

void MidiPlayerTK.MPTKEvent.Stop ()

Stop the note. DEPRECATED in V2. Replaced by MPTK_StopEvent in MidiStreamPlayer.

override string MidiPlayerTK.MPTKEvent.ToString ()

Build a string description of the Midi event.

Returns:

Member Data Documentation

long MidiPlayerTK.MPTKEvent.Track

Index of track.

long MidiPlayerTK.MPTKEvent.Tick

Time in Midi Tick (part of a Beat) of the Event since the start of playing the midi file. This time is independant of the Tempo or Speed. Not used for MidiStreamPlayer.

MPTKCommand MidiPlayerTK.MPTKEvent.Command

Midi Command code. Defined the type of message (Note On, Control Change, Patch Change...)

MPTKController MidiPlayerTK.MPTKEvent.Controller

Controller code. When the Command is ControlChange, contains the code fo the controller to change (Modulation, Pan, Bank Select ...). Value will contains the value of the controller.

MPTKMeta MidiPlayerTK.MPTKEvent.Meta

MetaEvent Code. When the Command is MetaEvent, contains the code of the meta event (Lyric, TimeSignature, ...). . Info will contains the value of the meta.

string MidiPlayerTK.MPTKEvent.Info

Information hold by textual meta event when Command=MetaEvent

int MidiPlayerTK.MPTKEvent.Value

Contains a value between 0 and 127 in relation with the Command. For:

- `Command` = NoteOn then Value contains midi note
- `Command` = ControlChange then Value contains controller value
- `Command` = PatchChange then Value contains patch value

int MidiPlayerTK.MPTKEvent.Channel

Midi channel fom 0 to 15 (9 for drum)

int MidiPlayerTK.MPTKEvent.Velocity

Velocity between 0 and 127

long MidiPlayerTK.MPTKEvent.Duration

Duration of the note in millisecond

int MidiPlayerTK.MPTKEvent.Length

Duration of the note in Midi Tick. MidiFilePlayer.MPTK_NoteLength can be used to convert this duration. Not used for MidiStreamPlayer. https://en.wikipedia.org/wiki/Note_value

List<fluid_voice> MidiPlayerTK.MPTKEvent.Voices

List of voices associated to this Event for playing a NoteOn event.

MidiPlayerTK.MPTKListItem

A list of string with index: midi, preset, bank, drum, ...

Public Attributes

- int Index
  Index in the list:
- string Label
  Label

Detailed Description

A list of string with index: midi, preset, bank, drum, ...

Member Data Documentation

int MidiPlayerTK.MPTKListItem.Index

Index in the list:

- `patch` num if patch list,
- `bank` number if bank list,
- `index` in list for midi.

string MidiPlayerTK.MPTKListItem.Label

Label

MidiPlayerTK.TrackMidiEvent

Midi event list (NAUdio format)

Detailed Description

Midi event list (NAUdio format)

Index

INDEX